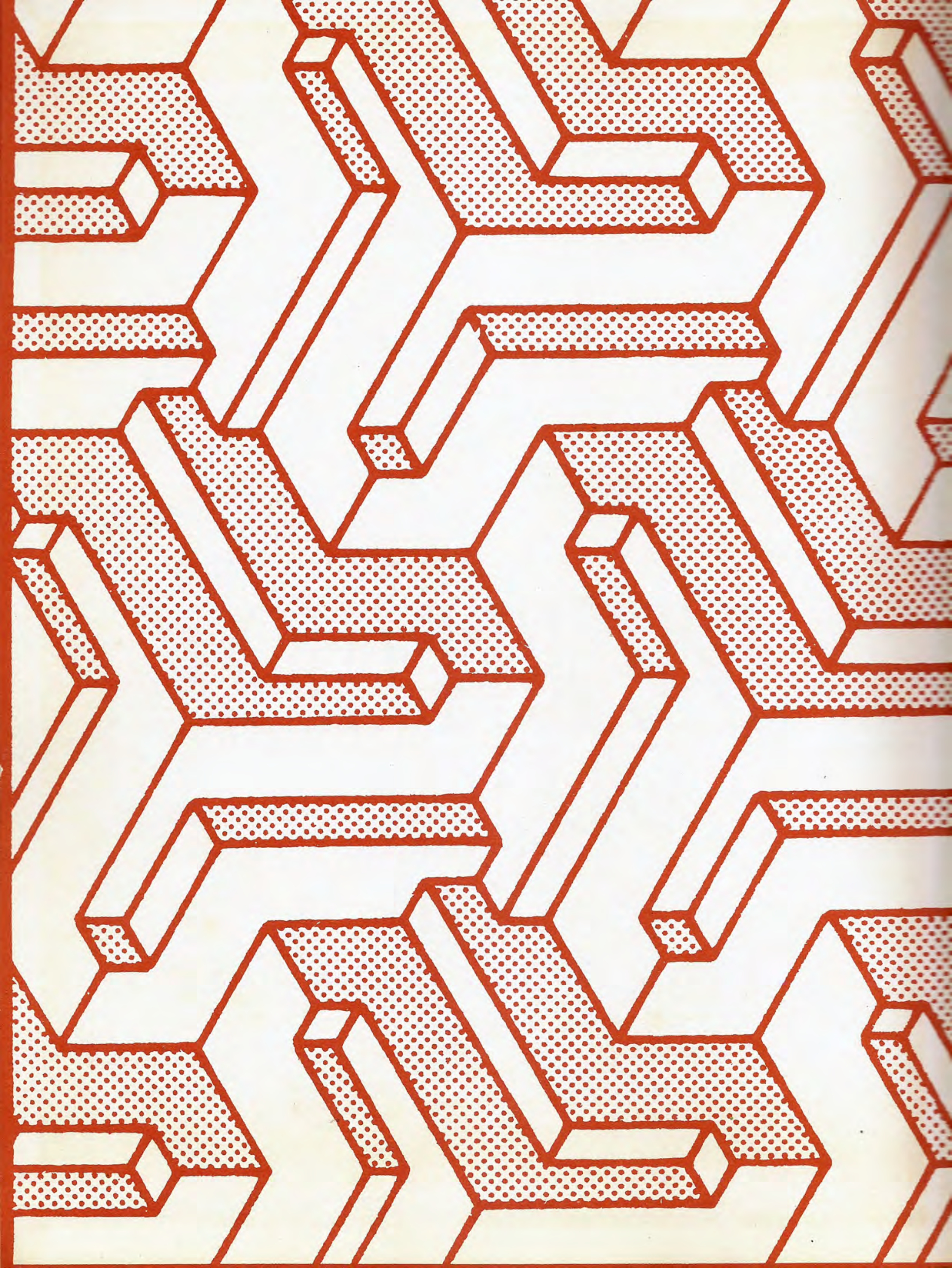
GRAN ENCICLOPEDIA INFORMATICA



SISTEMAS OPERATIVOS/2

EDICIONES NUEVA LENTE



GRAN ENCICLOPEDIA INFORMATICA

EDICIONES NUEVA LENTE



DISC 2, GEM™ STARTUP v2.0U.
© 1986 Amstrad Consumer Electronics plc.
Programs
© Copyright Digital Research Inc. 1985/86. All rights reserved.
© Microsoft® Corporation, 1981-85. All rights reserved.
46002
DISC 3, GEM DESKTOP™ v2.0U. / Locomotive BASIC 2™ v1.0
© 1986 Amstrad Consumer Electronics plc.
Programs
© Copyright Digital Research Inc. 1985/86. All rights reserved.
© Microsoft® Corporation, 1981-85. All rights reserved.
© 1986 Locomotive Software Ltd.
46003
DISC 4, DOS Plus™ STARTUP & Utilities v1.2 / GEM PAINT™ v1.0U
© 1986 Amstrad Consumer Electronics plc.
Programs
© Copyright Digital Research Inc. 1985/86. All rights reserved.
DISC 1, MICROSOFT® MS DOS® OPERATING SYSTEM
START UP & UTILITIES v3.2
© 1986 Amstrad Consumer Electronics plc.
© Microsoft® Corporation, 1981-85. All rights reserved.
46001
SISTEMAS OPERATIVOS

SUMARIO

Una publicación:

Ediciones Nueva Lente, S. A.

Director editor: MIGUEL J. GOÑI

Director de producción: SANTOS ROBLES.

Director de la obra: FRANCISCO LARA.

Colaboradores: PL/3 - MANUEL MUÑOZ - ANGEL MARTINEZ - MIGUEL DE ROSENDO - DAVID SANTAOLALLA - SANTIAGO RUIZ - LUIS COCA - MIGUEL ANGEL VILA - MIGUEL ANGEL SANCHEZ VICENTE ROBLES.

Diseño: BRAVO/LOFISH.

Maquetación: JUAN JOSE DIAZ SANCHEZ.

Ilustración: JOSE OCHOA.

Fotografía: (Equipo Gálata) ALBINO LOPEZ y EDUARDO AGUDELO.

Ediciones Nueva Lente, S. A.:
Dirección y Administración:
Benito Castro, 12. 28028 Madrid. Tel.: 245 45 98.

Números atrasados y suscripciones:
Ediciones Ingelek, S. A.
Plaza de la Rep. Ecuador, 2 - 1.°. 28016 Madrid.
Tel.: 250 58 20.

Plan general de la obra:
18 tomos monográficos de aparición quincenal.

Distribución en España:
COEDIS, S. A. Valencia, 245. Tel.: 215 70 97.
08007 Barcelona.

Delegación en Madrid:
Serrano, 165. Tel.: 411 11 48.

Distribución en Argentina:
Capital: AYERBE
Interior: DGP

Distribución en Chile: Alfa Ltda.

Distribución en México:
INTERMEX, S. A.
Lucio Blanco, 435
México D.F.

Distribución en Uruguay:
Ledian, S. A.

Edita para Chile:
PYESA
Doctor Barros Borgoño, 123
Santiago de Chile

Importador exclusivo Cono Sur:
CADE, SRL. Pasaje Sud América, 1532.
Tel.: 21 24 64. Buenos Aires - 1.290. Argentina.

Fotomecánica: Ochoa, S. A.
Miguel Yuste, 32. 28037 Madrid.

Impresión: Gráficas Reunidas, S. A.
Avda. de Aragón, 56. 28027 Madrid.

ISBN de la obra: 84-7534-184-5.
ISBN del tomo 16: 84-7534-247-7
Printed in Spain
Depósito legal: M. 27.605-1986

Precio de venta al público en Canarias, Ceuta y Melilla: 940 ptas.
Abril-1987

MS-DOS (1)

El más popular de los sistemas operativos para equipos de 16 bits



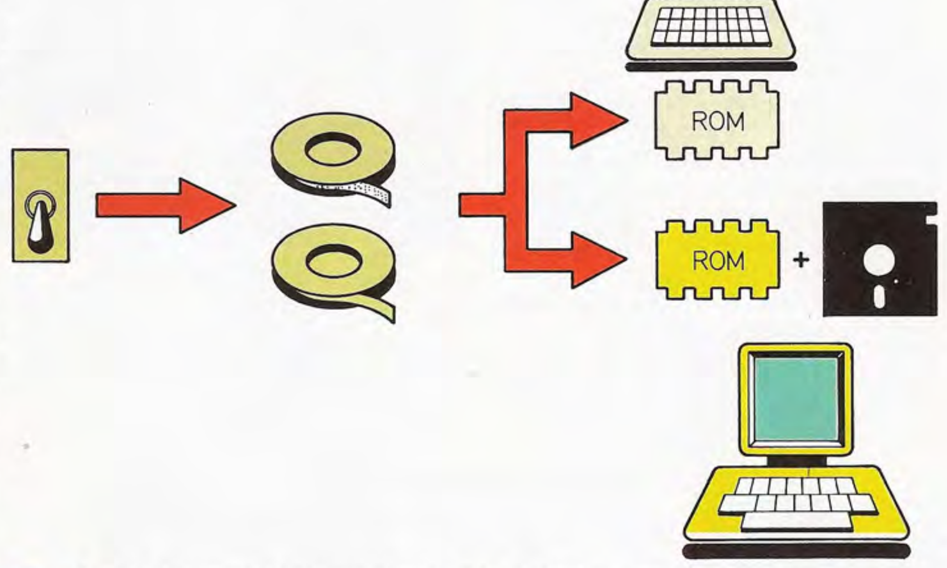
Son muy diversas las definiciones más o menos coloquiales y figurativas, aplicadas al sistema operativo: «la inteligencia elemental del ordenador» o «la mano invisible que guía el funcionamiento de la máquina»... Semejantes calificativos se aplican de forma genérica al conjunto de los sistemas operativos. Una familia constituida por una ingente variedad de miembros con mayor o menor presencia en el terreno informático.

¿Cuál es el motivo de la actual proliferación de distintos sistemas operativos destinados a microordenadores? La respuesta no es simple. En principio hay que considerar dos aspectos básicos que protagonizan tal dispersión. Por un lado la potencia y capacidad del microprocesador que rige el funcionamiento coordinado de cada ordenador; y por otro, la estrategia de la compañía fabricante del ordenador en lo relativo al tipo de aplicaciones y personas que van a utilizar sus productos.

Algunas consideraciones de tipo general contribuirán a vislumbrar los orígenes del sistema operativo que nos ocupa, el MS/DOS, y permitirán su comparación con otros sistemas operativos de su categoría. En los primeros años de la década de los 70, cuando la microinformática se encontraba aún en sus albores, los equipos estaban desprovistos de sistema operativo. La introducción de datos e instrucciones se realizaba por medios rudimentarios (a veces, por medio de simples interrupciones), y bajo la total supervisión y control del propio usuario.

Este procedimiento, largo, tedioso y, sobre todo, propenso a la introdución de errores, era obvio que no podía prolongarse en el tiempo. Así, aparecieron rápidamente sistemas operativos de gran simplicidad, los cuales se almacenaban en memoria permanente (ROM) y que eran, en realidad, poco más que lo que hoy denominaríamos cargadores o «loaders». Estos programas permitían que el ordenador leyera de una cinta perforada —y años más tarde de una cinta magnética—, un sistema operativo mucho más complejo. Esta idea, perfeccionada y refinada por una serie de consideraciones de tipo práctico, se ha adueñeado del campo de los ordenadores personales, tanto para uso doméstico como profesional.

En el campo de los microordenadores para uso doméstico suele ser frecuente una recesión del sistema operativo a la memoria permanente de la máquina (ROM). Al ser normalmente escasa la cantidad de memoria disponible del equipo, no resulta muy atrayente la idea de ocupar gran parte de la misma con un sistema operativo potente y dejan tan sólo una mínima porción de memoria para los programas del usuario. Con tal limitación, el equipo perdería gran par-



Con el nacimiento y progresiva evolución de los sistemas operativos, la introducción de los datos en el ordenador, así como el almacenamiento del propio sistema operativo, han sufrido una total modificación.



El éxito del MS-DOS se debe, en gran medida, a su adopción como sistema operativo del ordenador personal de IBM. En este equipo, las siglas MS-DOS se ven reemplazadas por las de PC-DOS.

te de su capacidad de trabajo. En este ámbito, cada fabricante suele crear su propio sistema operativo que, al ser desarrollado por el mismo, logra un más eficaz aprovechamiento de los recursos hardware del ordenador.

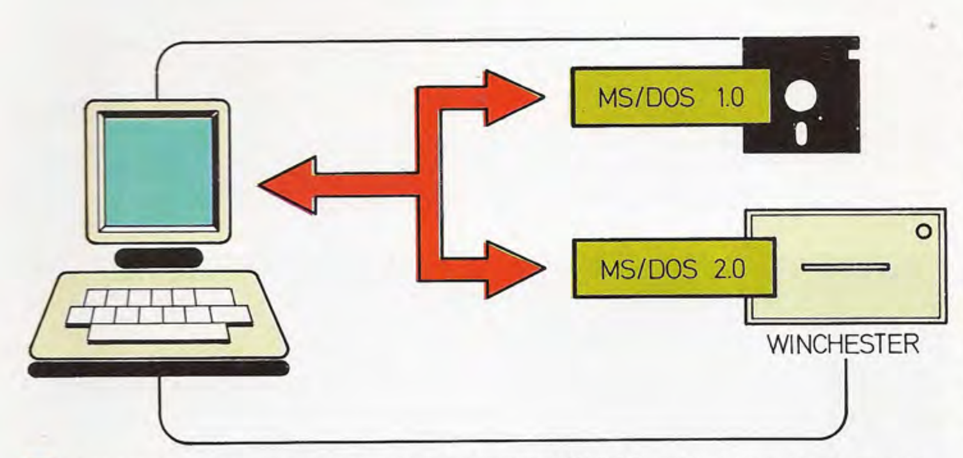


Al igual que el CP/M es el sistema operativo líder en el ámbito de los equipos de 8 bits, el MS-DOS es el sistema operativo imperante en el terreno de los equipos basados en un microprocesador de 16 bits. A la sombra del IBM-PC, han florecido un elevado número de ordenadores compatibles que día a día consolidan al MS-DOS en su liderazgo.

En el terreno de los ordenadores personales de tipo profesional o de gestión, la idea primitiva de cargar en la máquina un sistema operativo más potente desde un medio de almacenamiento externo, es la que se ha consumado. Además, se manifiesta una acusada tendencia a que distintos equipos utilicen sistemas operativos comunes.

Los motivos que han empujado esta vía de evolución son, principalmente, tres. En primer lugar, la mayor capacidad de memoria de las nuevas máquinas permite destinar un cierto espacio al sistema operativo, sin por ello minimizar la memoria reservada al usuario.

Por otra parte, para el fabricante del ordenador, la creación de un sistema operativo competitivo y que haga pleno empleo de la capacidad de su equipo es una tarea larga y económicamente onerosa. A título ilustrativo, basta considerar que el S.O. empleado en el ordenador LISA, de la firma Apple Computers, necesitó un total de 200 años-programador para su desarrollo. Por lo demás, no hay que perder de vista la labor ingente necesaria para adaptar programas de aplicación estandarizados a un nuevo sistema operativo.



Las sucesivas versiones del sistema operativo MS-DOS están orientadas al manejo de diversas unidades para el almacenamiento externo de información. La versión DOS 2.0 completa su capacidad de trabajo sobre discos flexibles con la facultad de controlar unidades de disco rígido.

El tercer motivo, y quizá el más importante, reside en consideraciones de mercado. Hoy en día, con un mercado de microordenadores de tipo personal en constante crecimiento, y extendiéndose desde el ámbito doméstico hasta el empresarial, no parece muy consecuente la idea de que cada fabricante desarrolle su propio sistema operativo. La baza de las aplicaciones estandarizadas, creadas para su compatibilidad con los sistemas operativos de mayor resonancia queda fuera de semejante alternativa. Hay que tener en cuenta que muchos ordenadores no son expertos en informática y programación; su único deseo es que el ordenador les libere de tareas rutinarias a través normalmente de programas plenamente contrastados. En efecto, los paquetes de aplicación de mayor éxito comercial —por ejemplo, las hojas electrónicas o procesadores de texto— han sido desarrollados por compañías de software independientes; el hecho de que no puedan ser empleados en su propio ordenador opera en contra del fabricante.

Los orígenes del MS/DOS

El sistema operativo MS/DOS es un producto creado por la firma americana Microsoft y destinado a ordenadores basados en los microprocesadorse de 16 bits INTEL 8088 o INTEL 8086. Este sistema operativo se ha convertido en un verdadero estándar en el terreno de los equipos de 16 bits; al igual que el sistema operativo CP/M lo es para los ordenadores de 8 bits.

El éxito del MS/DOS tiene su raíz en el hecho de que la multinacional IBM lo eligiera como el sistema operativo de su

ordenador personal: el IBM-PC. La gran aceptación del IBM-PC ha empujado al MS/DOS a la posición de liderazgo que hoy ocupa.

El camino que llevó a esta decisión es simple y directo e ilustra las anteriores consideraciones. La elección por IBM del microprocesador INTEL 8088 deriva, en parte, de la peculiar arquitectura de este microprocesador. Aunque éste opera internamente con palabras de 16 bits, la entrada y salida de datos se realiza en bloques de tamaño byte (8 bits). Las ventajas que de ello derivan son claras. Por un lado, la posibilidad de utilizar circuitos simples y baratos, adecuados para trabajar asociados a un bus de 8 bits. Además del hecho de que el repertorio de instrucciones en lenguaje ensamblador para el procesador 8088, es un conjunto ampliado de las utilizadas por el microprocesador de 8 bits 8080. Ello significa que un gran número de programas desarrollados para el 8080, sólo necesitan ser reensamblados para su traslado al microprocesador de 16 bits 8088. De esta forma, se accede, sin excesivas complicaciones, a un amplio abanico de programas de éxito comercial. Dado que el MS/DOS fue creado, precisamente, para gobernar al microprocesador 8088, la elección ofrecía pocas dudas.

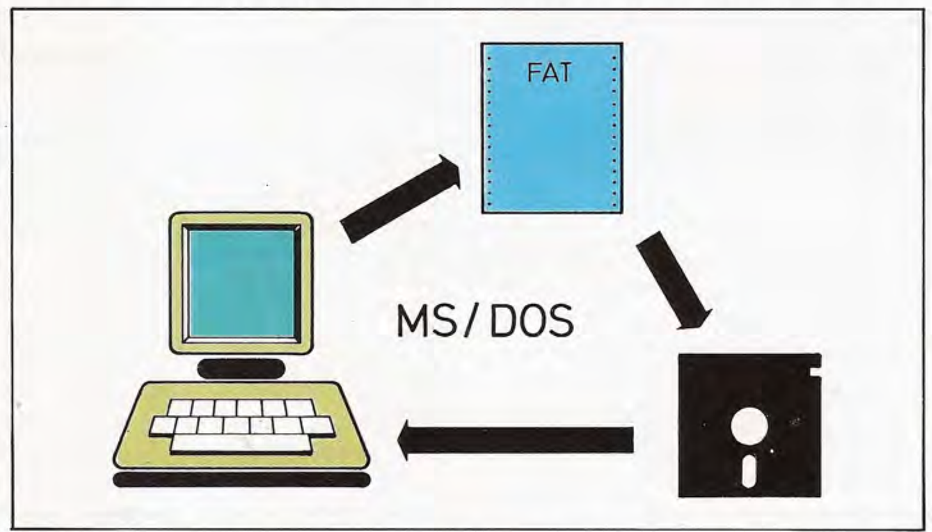
Conceptualmente, el MS/DOS guarda una gran similitud con el sistema operativo CP/M, ya que ha seguido sus pautas.

El MS/DOS está orientado esencialmente a la gestión de archivos en disco, de ahí su denominación: DOS (Disk Operating System). En su versión, DOS 1.0, está configurado para operar con dos unidades de disco flexible, mientras que en la versión DOS2.0 se emplea para el control de unidades de disco rígido de tecnología Winchester.

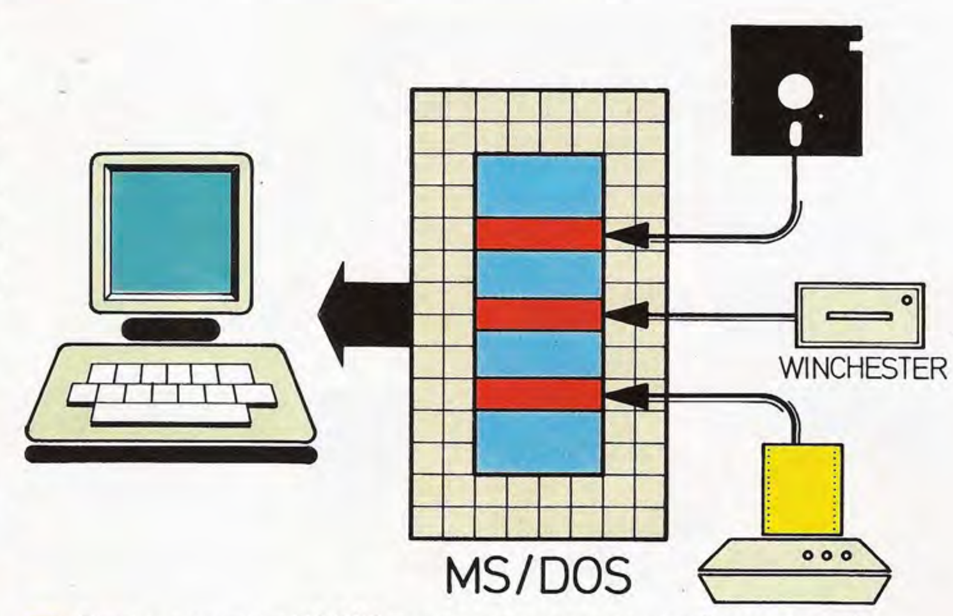
Una gran ventaja de este aspecto del MS/DOS, es que cada uno de los distintos periféricos conectados al ordenador está considerado en un módulo propio del sistema operativo. Con ello, es obvio que el equipo puede adaptarse al uso de muy diversas categorías de periféricos. Este método presenta, sin embargo, una desventaja clara: está pensado más en función de las necesidades del ordenador que de las del propio usuario, lo que obliga a éste a conocer los diferentes comandos del sistema operativo, su función y su formato exacto.

Comparación entre el MS/DOS y otros sistemas operativos

Desde luego, la orientación del MS/DOS hacia los equipos de 16 bits, marca una distinción activa respecto a los sistemaas operativos destinados a los microprocesadores de 8 bits. Sin embargo, el MS/DOS también presenta notables diferencias con otros sistemas operativos destinados a microprocesadores de 16 bits; tal es el caso del UNIX, un producto de los laboratorios Bell. También son considerables sus diferencias con los sistemas operativos integrados —«user-friendly»— que equipan a los ordenadores Lisa y Macintosh de la



Los procesos de carga de ficheros en la memoria central se ven acelerados por la presencia de una tabla que revela el emplazamiento de los mismos en la memoria de masa.



El sistema operativo MS-DOS considera por separado a los diversos tipos de periféricos conectables. Cada uno de ellos es controlado por un módulo distinto del sistema operativo.



Tal y como reflejan las siglas DOS (Disk Operating System), el MS-DOS está especialmente orientado a la gestión de archivos de disco.

firma Apple. Veamos algunas diferencias esenciales que presenta el MS/DOS respecto a otros sistemas operativos de gran aceptación.

- CP/M

Ambos sistemas operativos presentan una gran similitud a nivel general, aunque difieren en aspectos muy importantes. Básicamente, el CP/M se ve frenado por el reducido volumen de memoria que pone en sus manos un microprocesador de 8 bits. Ello se traduce en una disminución en el número de comandos a disposición del operador. La consecuencia práctica es que para llevar a cabo todas las funciones necesarias, es preciso completar cada comando con una serie de parámetros adicionales, con el consiguiente problema de memorización por parte del usuario. Asimismo, los requerimientos de memoria hacen que los mensajes de error sean crípticos y no muy numerosos.

Por contra, el MS/DOS ofrece un mayor número de comandos y con un formato más lógico, así como un repertorio de elaborados mensajes de error y subrutinas de comprobación. Una diferencia adicional es que el sistema operativo mantiene continuamente en memoria una tabla de localización de archivos (FAT). De esta forma, los accesos a los ficheros en disco son mucho más rápidos, al saber el sistema operativo donde tiene que posicionarse en el disco para localizar la información.

- UNIX

Este sistema operativo presenta la gran diferencia de estar escrito en un lenguaje de alto nivel, con lo cual es totalmente independiente del hardware. En consecuencia puede aplicarse a casi cualquier ordenador, desde personales hasta grandes equipos, con las consiguientes ventajas en la compatibilidad de programas. Otra diferencia sustancial reside en el hecho de que la interacción del usuario con el sistema operativo es posible canalizarla, de acuerdo con sus preferencias y necesidades, a través de menús. Ello simplifica en gran medida la tarea de dar órdenes a la máquina. El UNIX se desarrolló inicialmente para mini-ordenadores; su descenso al terreno de los microordenadores está condicionado por la característica apuntada al principio. En este ámbito, la eficacia del UNIX dependerá de la presencia en el equipo de un potente microprocesador que no ralentice los ciclos de respuesta; en este punto, el UNIX es un voraz consumidor de ciclos de máquina.

- S.Os integrados para Apple Lisa y MacIntosh

Ambos ordenadores comparten un sistema operativo análogo. Este es único en su categoría. Se trata de un sistema operativo totalmente orientado al usuario y en el que el diálogo se reduce al desplazamiento de un indicador sobre la pantalla, gobernado por el ya popular «ratón». Las operaciones y procesos se ordenan movilizando el indicador sobre un grupo de menús e iconos (gráficos representativos de ficheros, procesos y acciones) presentes en la pantalla.

En definitiva es la propia filosofía de estos sistemas operativos la que se distancia por completo de la propia del MS/DOS y de cualquier otro S.O. orientado a la máquina.

Por lo demás, hay que señalar que los referidos sistemas operativos para los modelos Apple Lisa y MacIntosh, operan sobre un microprocesador con arquitectura interna de 32 bits, aunque con estructura de comunicación externa de 16 bits: el Motorola 68000.

El DOS 1.0

La versión DOS1.0 del sistema operrativo MS/DOS está destinada a la gestión de archivos en disco flexible (disquete). Por contra, la versión DOS.2.0 más avanzada, se orienta a la gestión de discos rígidos. En la versión que ahora nos ocupa, la 1.0, el ordenador almacena todos los ficheros de datos bajo un directorio común. Ello equivale al hecho de guardar a la totalidad de los datos dentro de carpetas alojadas en un mismo cajón, sin disponer de otro elemento identificador que su nombre a la hora de proceder a cualquier operación de búsqueda.

La esencia de este sistema operativo reside, tal como se ha indicado, en la gestión de los archivos residentes en disco flexible. Las unidades de disco asociadas al equipo se identifican, de



Proceso de búsqueda de un comando por parte del sistema operativo MS-DOS.

forma consecutiva, por medio de las letras A,B,C... En el caso de la versión DOS 2.0 las unidades de disco rígido adoptan un criterio de identificación que se describirá en un capítulo posterior.

Debido a su propia estructura, este sistema operativo necesita la presencia de dos unidades de disco. Si sólo dispone de una unidad, el propio MS/DOS se encargará de simular la presencia virtual de la segunda unidad de disco. Ello no afectará al modo habitual de operación, si bien, el usuario se verá obligado a cambiar continuamente de disquete, con la consiguiente deceleración en la velocidad de trabajo.

De cara a la búsqueda de información en disquete, el sistema operativo debe conocer cuál es la unidad de disco en la que reside. Ello se indicará suministrando al DOS la clave de la unidad implicada. De omitir tal indicación, el sistema operativo toma por defecto una de ellas para buscar la información solicitada.

La referencia a la unidad implicada aparecerá en el «prompt» o «indicador» de presencia, quien revela que el ordenador está en disposición de recibir una orden:

A>unidad de disco flexible A
B>unidad de disco flexible B

Si se desea cambiar la unidad de disco tomada como opción por defecto, por ejemplo de A a B, bastará con comunicarlo al ordenador como sigue:

A>B: <CR>

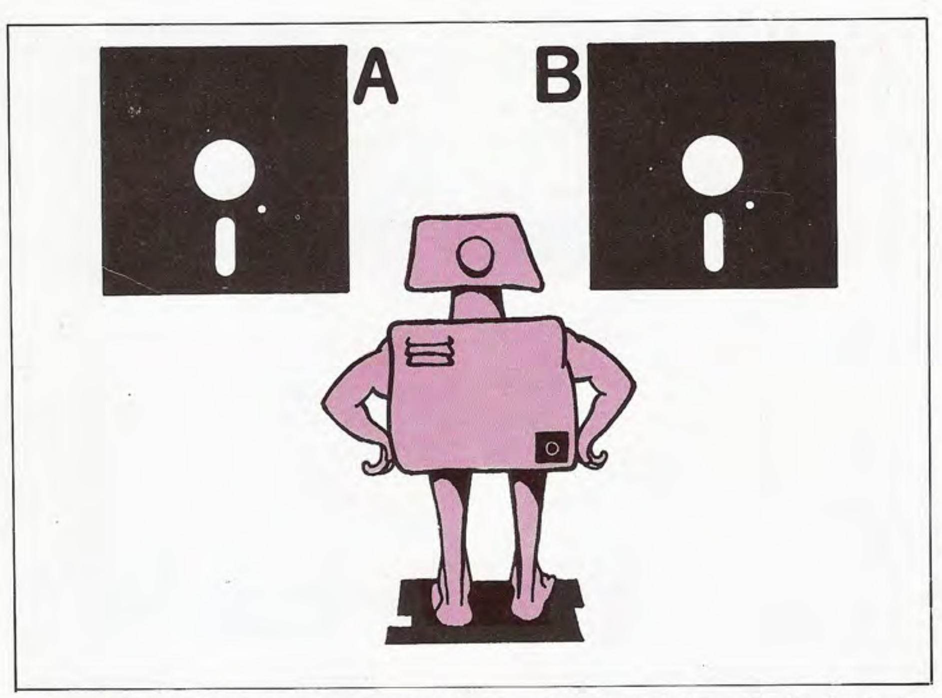
(CR= retroceso de carro; tecla RETURN o ENTER).

A partir de este preciso instante, la unidad de disco B será la que adopte el sistema por defecto. Su indicativo (B) pasará a sustituir a la letra A en el «prompt».

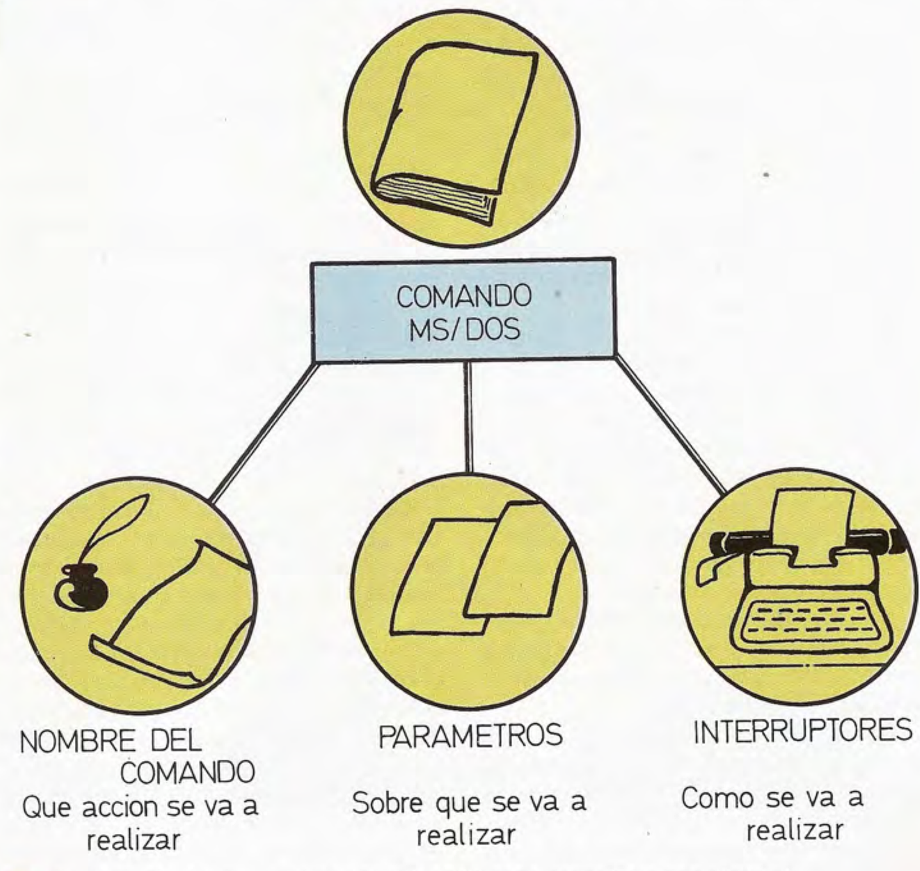
Tipos de comandos

Para que el ordenador sea capaz de acometer cualquier tarea, es preciso que reciba las órdenes adecuadas en forma de comandos.

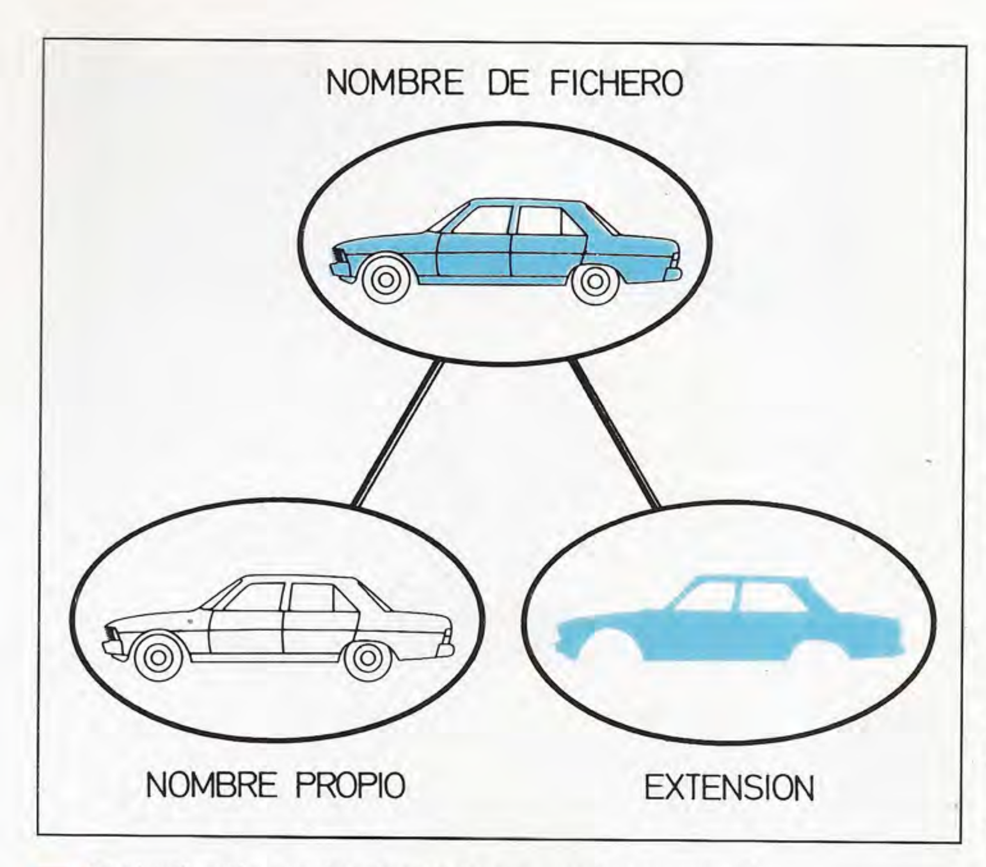
En esencia, los comandos del DOS SON, realmente, programas activados por la introducción de su respectivo nombre. Como tales programas, cabe



El MS-DOS puede leer y almacenar datos en cualquiera de las unidades de disco, sin más que especificarlo en los comandos que realicen tal cometido.



La formulación de un comando MS-DOS debe especificar con claridad todos los extremos relativos a la acción a ejecutar.



La identificación completa de cada fichero exige la presencia del nombre propio y del apellido o extensión. Este último revela el carácter de los datos contenidos dentro del fichero.

preguntarse cómo se almacenan y cómo son ejecutados por la máquina, teniendo en cuenta su disponibilidad ante los requerimientos del usuario.

Al igual que sucede con el sistema operativo CP/M, el MS/DOS incorpora toda una serie de comandos que residen habitualmente en la memoria central del equipo. Estos pasan a la memoria interna durante el proceso inicial de carga del sistema operativo. Dada su localización durante los periodos de funcionamiento del ordenador, estos comandos se denominan *internos*.

En este aspecto, el MS/DOS es superior al CP/M: posee un mayor número de comandos internos y éstos poseen un formato más lógico.

En todo caso, el número de comandos internos no puede crecer en exceso, dado que el ordenador tiene un limitado volumen de memoria primaria. Resulta obvio que cuantos más comandos residan en su interior, tanto menor será el espacio de memoria que quedará a disposición de los programas de usuario. En virtud de esta limitación, se ha llegado a un compromiso: una serie de comandos, denominados externos, permanecen almacenados en el disquete que contiene el sistema opeativo, concretamente en ficheros de tipo COM. Cada vez que es necesaria su presencia, éstos son cargados en la memoria interna como paso previo a su ejecución.

Bajo tal perspectiva, cuando se invoca a un comando MS/DOS pueden suceder dos cosas. Si el comando es interno, éste se ejecutará inmediatamente; por el contrario, si es externo, el sistema operativo no lo encontrará en su lista de comandos internos y, en consecuencia, procederá a buscarlo en el disquete que se encuentre en la unidad considerada por defecto.

Este aparecerá en forma de fichero de tipo COM. Una vez localizado, lo trasladará a la memoria central y procederá a ejecutarlo. Desde luego, todo este proceso supone el consiguiente retardo.

Al realizar semejante operación, el ordenador puede presentar en la pantalla un mensaje de error; mensaje que puede obedecer a dos motivos:

— Se ha escrito de forma incorrecta el nombre del comando.

— Aun siendo correcta su denominación, se trata de un comando externo cuyo respectivo fichero no se encuentra almacenado en el disco en el que se produce la búsqueda.

Notación de los comandos del DOS

El MS/DOS es un sistema operativo orientado principalmente al ordenador (como contrapartida a los sistemas operativos orientados al usuario). De ahí que exija al usuario el aprendizaje de los distintos comandos utilizables, así como de su formato.

El formato genérico de los comandos del MS/DOS adopta la siguiente disposición:

>NOMBRE DEL COMANDO PARAMETROS

Cada una de ambas zonas realiza una función dentro de la orden.

— Nombre del comando: indica la acción a realizar.

— Parámetros: definen sobre qué va a actuar el comando y cómo va a realizar dicha tarea. Los parámetros que se ocupan de este segundo cometido se denominan interruptores (switches); normalmente coinciden con los símbolos «/» o «–» dependiendo del equipo.

Un ejemplo práctico lo constituyen los siguientes comandos, adecuados para copiar archivos.

>COPY GEI.TEX RESUMEN.TEX
>COPY A:*.*B

El primero de ellos copiará el fichero GEI.TEX en otro cuya denominación es RESUMEN.TEX. Al no indicarse el disco afectado, se da por supuesto que ambos ficheros pertenecen al disco insertado en la unidad que el sistema toma por defecto en ese preciso instante. El indica-

tivo de la mencionada unidad aparecerá en el «prompt» delante del símbolo «>».

El segundo comando incluye de nuevo dos parámetros: uno para especificar cuál es la información a copiar (todos los archivos del disco A) y otro para indicar cuál es el destino de la copia (el disco B). Como respuesta, el ordenador copiará a todos los ficheros del disco A en el disco situado en la unidad B, otorgando a los ficheros de destino el mismo nombre que poseen sus respectivos ficheros originales.

La sintaxis de este segundo comando presenta algunas particularidades que posteriormente se analizarán con mayor detalle. Tal es el caso, por ejemplo, del uso de los asteriscos al definir los ficheros a copiar. Por el momento es suficiente con saber que el parámetro «A:*.*», alude a todos los ficheros almacenados en el disco A.

En ambos ejemplos, la separación entre los distintos comandos coincide con un espacio en blanco. Algunos equipos admiten otros caracteres, además del espacio en blanco, como separadores de parámetro: coma (,), punto (.), punto y coma (;)...

Los ficheros y su denominación

Los programas, datos y comandos externos del sistema operativo MS/DOS se almacenan en disco adoptando la forma de ficheros. Para identificarlos unívocamente, cada uno de ellos recibe un nombre. Este ha de ser forzosamente distinto para cada uno de los ficheros, en versión DOS.1.0.

En la versión 2.0, que se estudiará en un próximo capítulo, es posible encontrar ficheros con el mismo nombre; si bien, el proceso de creación de los mismos es tal que permite diferenciarlos uno de otro.

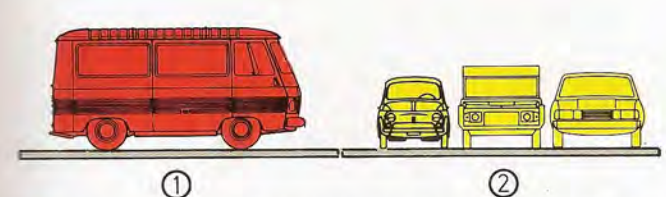
La identificación completa de un fichero consta de dos zonas, separadas entre sí por un punto. La primera zona es imperativa, mientras que la segunda es opcional. Ambas zonas se denominan, respectivamente, *nombre* propio del fichero («proper») y *extensión* del nombre del fichero. El propio o «proper» debe estar compuesto por al menos un carácter, y su longitud máxima puede elevarse a 8 caracteres. Estos pueden ser letras, números o determinados signos de puntuación que no se utilicen en la formulación de comandos. De ahí que no sea admisible la inclusión de puntos ni espacios en blanco dentro del nombre. Algunos nombres correctos de ficheros son los siguientes: DATOS,A7, BXY23 o ABC-34.

La extensión del nombre del fichero es un apéndice que consta de un máximo de tres caracteres. Su mayor utilidad estriba en que pueda dar una idea del tipo de información que contiene el fichero. Por ejemplo, los comandos externos del MS/DOS residen en ficheros cuya extensión es «COM» (extensión indicativa de comando).

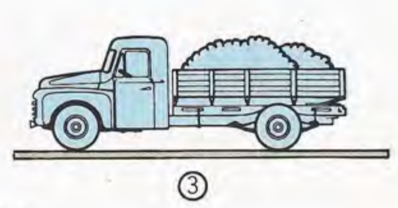
Cabe encontrar también una categoría de ficheros, denominados de tipo batch, en los que el usuario almacena un conjunto de comandos del DOS. Los referidos comandos se ejecutarán secuencialmente al invocar al fichero en cuestión, de igual forma que si el usuario los fuera introduciendo sucesivamente a través del teclado. Este tipo de ficheros, cuyo efecto práctico es similar al de los ficheros que acogen a los comandos externos, incluyen como extensión del nombre propio la expresión «BAT».

Un dato a señalar relativo a la denominación que identifica a los ficheros, es la posibilidad de utilizar *referencias ambiguas* o «wildcards». Su interés estriba en que permite extender la actuación de un comando a un grupo genérico de ficheros, sin necesidad de especificar el nombre de cada uno de ellos.

En el sistema operativo DOS, la referencia ambigua adecuada para ocupar la zona de nombre o de extensión del nombre es el símbolo «*» (asterisco). Un ejemplo práctico lo constituye la orden COPY formulada en el apartado precedente. El empleo de «*.*» como sustitutivo de la denominación del fichero, hará que sean copiados todos los ficheros del disco origen. Si la designación del fichero coincide con «*. COM», los archivos afectados por la operación de copia serán todos aquellos cuya extensión sea COM, con independencia de su nombre específico.







Por medio de las referencias ambiguas o «wildcards» es posible agrupar ficheros que presenten características comunes, de tal forma que el comando a ejecutar actúe sobre todos ellos.

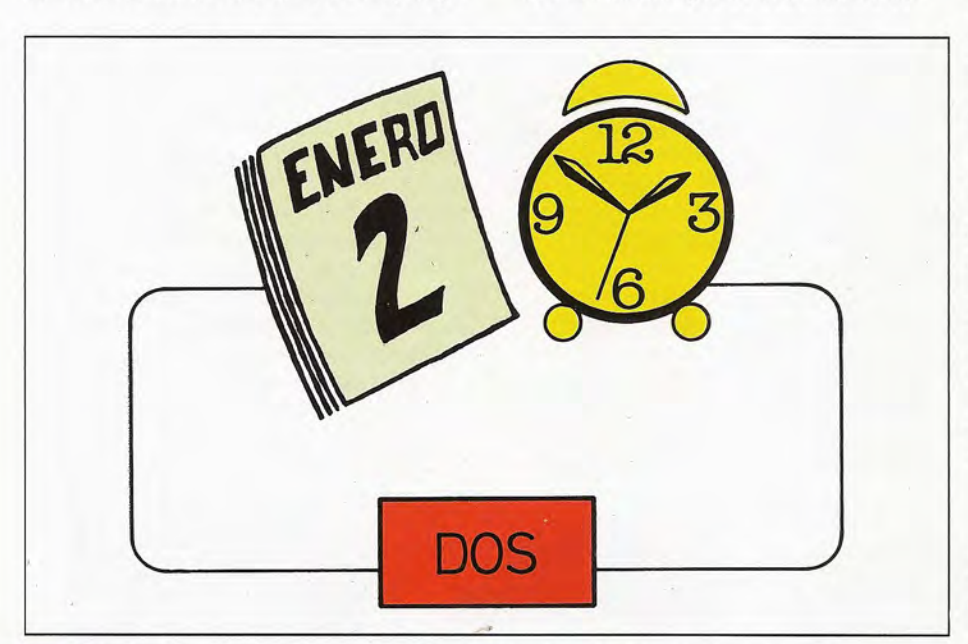


Normalmente, los equipos que incorporan el sistema operativo MS-DOS suelen disponer de una doble unidad de disco. En la fotografía aparece el Ericsson PC, un ordenador personal compatible con el IBM-PC y equipado con una doble unidad para disco flexibles.

Una vez introducida la filosofía general del MS/DOS y detallados los aspectos básicos de la versión DOS1.0, llega el momento de presentar los comandos que brinda este sistema operativo.

Los comandos que se describen a continuación son los más comunes y de uso más frecuente. Posteriormente, se acometerá el estudio de los restantes comandos, adecuados para ejecutar tareas más complejas.

Los comandos que van a ser objeto de análisis en los próximos párrafos, son comunes a las versiones 1.0 y 2.0 del sistema operativo MS/DOS. A pesar de su analogía, cabe señalar que en la versión más avanzada (DOS 2.0), éstos tienen mayor potencia en algunos casos. Tal circunstancia quedará reflejada en el texto. Un detalle a precisar de antemano es que cada fabricante, aun adoptando la pauta general, introduce pequeñas diferencias en los nombres y, en ocasiones, en la sintaxis de ciertos comandos. Ello exige un repaso del manual de usuario específico del ordenador que vaya a utilizarse.



El MS-DOS cuenta con un reloj interno cuya función básica es suministrar información al sistema. Su manipulación se realiza por medio de los comandos DATE y TIME.

Comandos de control de tiempo

El sistema operativo MS/DOS ofrece al usuario la posibilidad de operar con un reloj interno. La función básica de dicho reloj es suministrar información al sistema operativo, de tal forma que cuando se crea o actualiza algún tipo de información sobre un disco, queda constancia del instante en el que se realizó esta operación. La referida indicación es sumamente importante, toda vez que suministra al usuario una completa información acerca de cuándo fue la última vez que se actualizaron ciertos datos. Ello le permitirá conocer sobre qué información se trabajó y en qué periodo de tiempo exactamente. Sin lugar a dudas, dicha posibilidad facilita el control sobre el uso de datos. Los comandos que permiten acceder al reloj interno son DATE y TIME, cuyas respectivas funciones son:

DATE: Presentar y actualizar la fecha
TIME: Presentar y actualizar la hora

La presentación de la fecha y la hora parece una tarea bastante simple, y de hecho es así. La posilidad de actualización deriva de que el reloj interno es volátil y, en consecuencia, ha de ser actualizado cada vez que se conecta al ordenador. Tras finalizar el proceso de arranque en frío y antes de iniciar el pro-

ceso de operación normal, el propio sistema operativo interroga al usuario respecto a la fecha y hora actuales. Mientras el ordenador no sea desconectado, el reloj seguirá funcionando y será capaz de pasar de un día a otro, así como de detectar si el año en curso es bisiesto con el fin de introducir las oportunas correcciones.

Para operar con ambos comandos, caben dos modalidades:

— Tecleando el nombre del comando y pulsando la tecla de retroceso de carro (RETURN, ENTER o <CR>).

En este caso, el ordenador presenta en pantalla la fecha y hora actuales, y aguarda a su posible modificación. Si se pulsa <CR> directamente, el dato no sufrirá alteración alguna. Por el contrario, de introducir una nueva fecha u hora, ésta será almacenada en memoria al accionar <CR> y el ordenador empezará a contar el tiempo desde ese instante.

— Tecleando el nombre del comando y el dato de fecha u hora correspondiente.

En esta modalidad se produce la actualización de forma indirecta, sin realizar la máquina ningún tipo de consulta. Por ejemplo:

DATE 23-01-84 <CR>

Comandos de preparación de disco

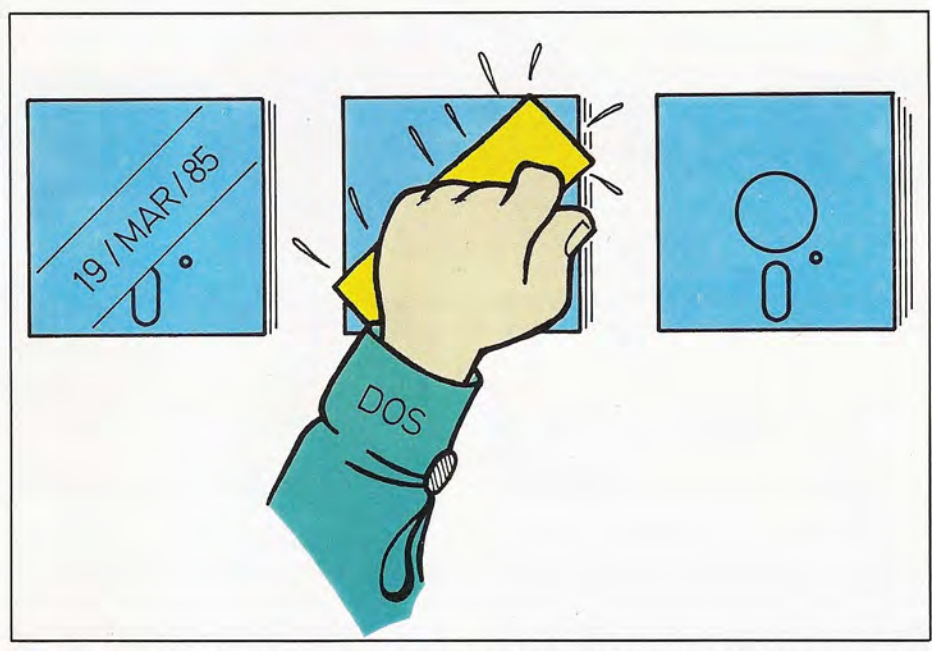
Dada su orientación a la actividad de gestionar archivos en disco, el MS/DOS pone a disposición del usuario una serie de herramientas destinadas a la preparación de los disquetes. Estas adoptan la forma de los comandos externos FORMAT, SYS y DISKCOPY.

FORMAT

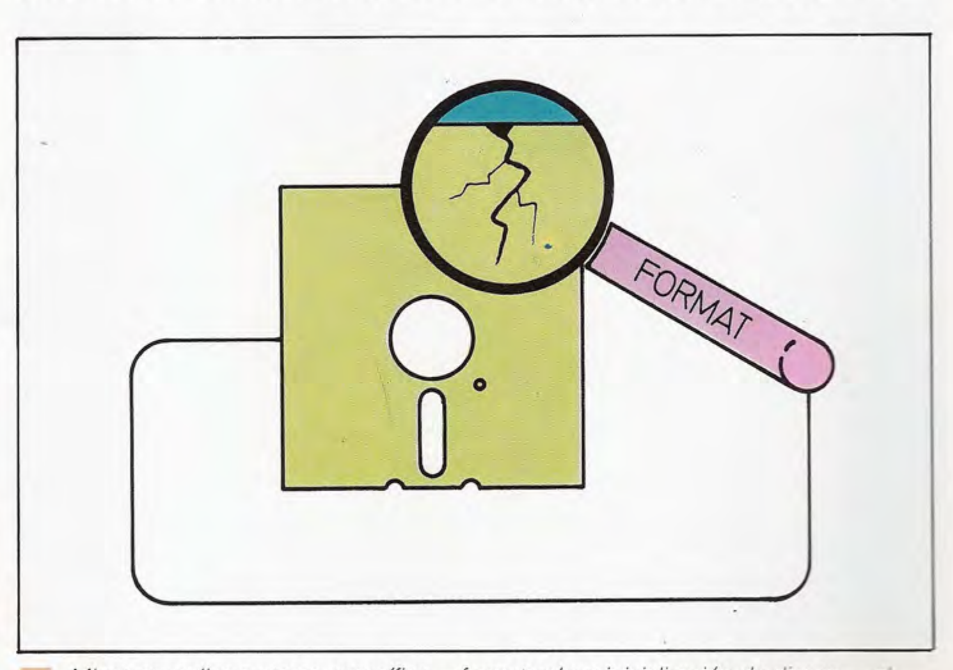
La misión del comando FORMAT es la de inicializar (formatear) un disquete virgen. Sobre éste crea una virtual «guía electrónica» que permitirá al sistema operativo trabajar posteriormente con el mismo. Durante esta operación, el propio comando examina el disquete, indicando los posibles defectos que este pudiera tener. En el caso de existir algún defecto, FORMAT lleva a cabo dos funciones.

— Advertir al usuario de la presencia de los defectos indicando el tamaño en bytes de la zona defectuosa, así como la cantidad total de memoria utilizable en el disco.

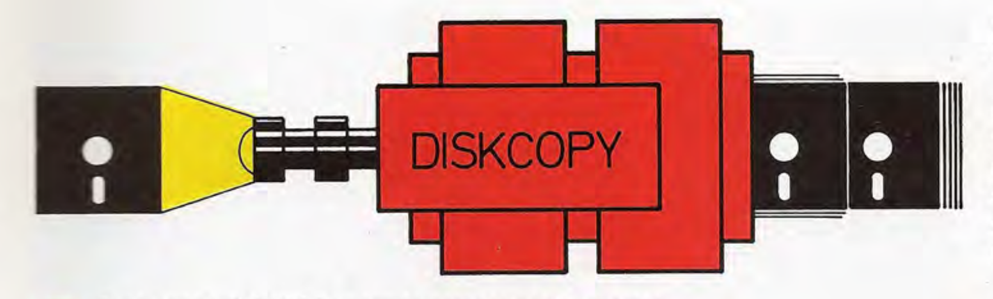
— Marcar el área defectuosa por medio de unos indicadores. Esta no será utilizable, aunque sí será posible utili-



Cada vez que se crea o actualiza el contenido de un fichero, el sistema operativo deja constancia del instante en el que se realizó la operación; para ello cuenta con el apoyo que le ofrece el reloj interno.



Mientras realiza su tarea específica —formateado o inicialización de discos—, el comando FORMAT examina el estado del disco y señala las zonas defectuosas del mismo.



El comando DISKCOPY permite obtener copias exactas de la información almacenada en un disquete.

zar el resto del disco para almacenar datos.

Lo normal es que el disco se formatee sin mayores problemas. De no ser así, y si el número de defectos fuera apreciable, será preciso rechazar el disco en cuestión. Posiblemente el disquete sea de fabricación defectuosa, o bien puede exitir algún problema con la cabeza grabadora de la unidad de disco.

Una posibilidad que permite el comando FORMAT es la de incluir los ficheros que contiene el sistema operativo en el disquete que se está formateando. Con ello, el nuevo disco podrá utilizarse para el arranque en frío del ordenador sin necesidad de recurrir a un disquete especial. Tal posibilidad la abren los interruptores «S» o «—S», dependiendo del modelo de ordenador utilizado.

Otros interruptores permiten tener en cuenta el tipo de disco en uso (si es de simple o doble densidad) en orden a realizar un formateado correcto.

FORMAT es un comando cuyo empleo puede resultar peligroso, toda vez que puede actuar sin problema alguno sobre un disco útil, repleto de datos. El resultado será la pérdida de todos ellos. En consecuencia, su puesta en práctica exige la adopción de las debidas precauciones.

SYS

El comando SYS se encarga de transferir el sistema operativo DOS a un disquete ya formateado. La necesidad de ejecutarlo puede responder a uno de los siguientes motivos:

— El uso de un programa que exige la presencia de una copia del sistema operativo en el disco que almacena el programa.

— La aparición de una nueva versión del DOS. Este comando permitirá introducirla en aquellos disquetes que contengan la versión antigua, sin dañar los ficheros de datos que pudieran estar almacenados.

Hay que precisar que este comando no realiza la copia de un fichero del sistema operativo denominado COMMAND.COM. Este debe copiarse de forma independiente, por medio de la orden COPY.

DISKCOPY

Su cometido es copiar el contenido de un disco íntegramente en otro, sin ningún tipo de operación intermedia. El resultado es una reproducción exacta del disquete original. Hay que hacer notar, sin embargo, que en algunas versiones del sistema operativo este comando no reproduce el formateado; de ahí que sea necesario inicializar previamente el disco que va a recibir la copia.

Cuando se desea transferir íntegramente los datos de un disco a otro y el disquete de origen se encuentra totalmente lleno, este comando se revela como sumamente eficaz. A pesar de su gran efectividad, hay algunas precisiones que conviene señalar:

— El comando DISKCOPY no admite en ninguno de los dos discos implicados en la operación de copia la existencia de algún defecto de los que el comando FORMAT detecta y señala.

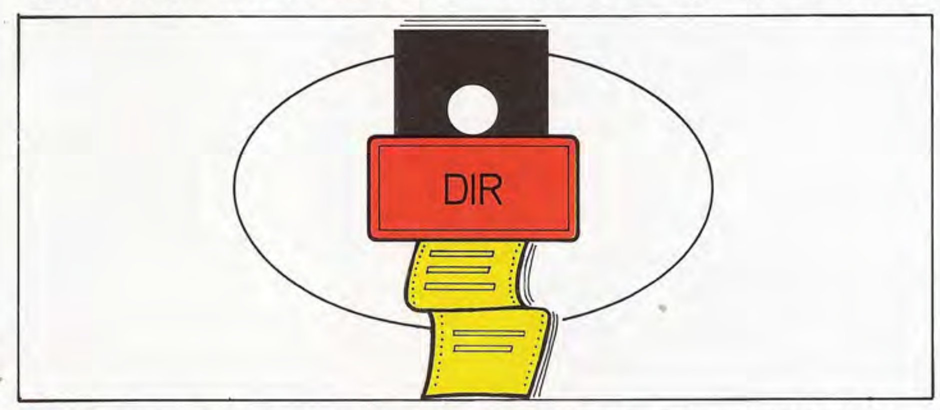
— No optimiza los resultados del disco. En consecuencia, si en el disquete original hay pocos datos y están distribuidos esporádicamente en su superficie, esta misma distribución será la que aparezca en la copia.

Comandos para la administración del disco

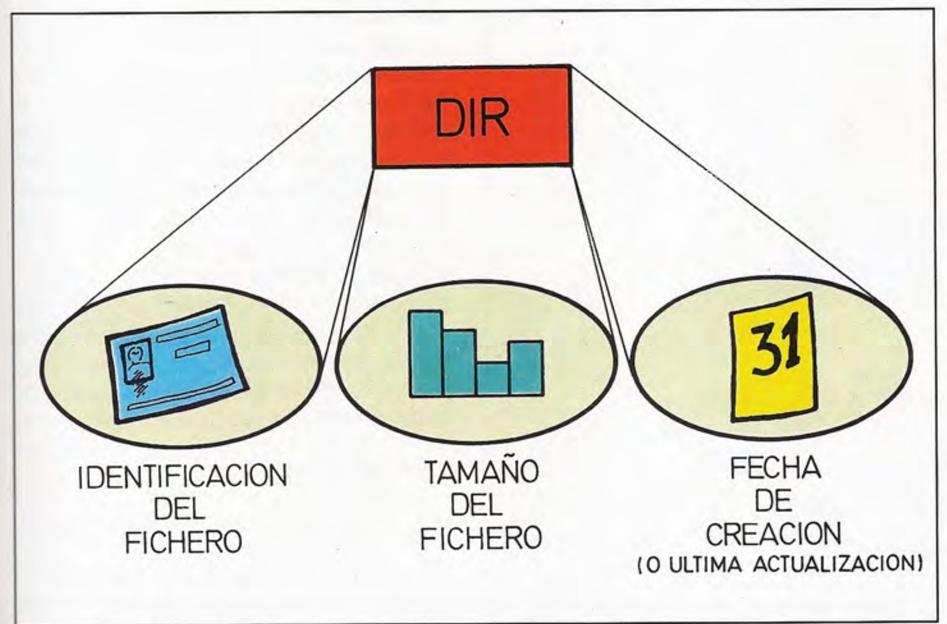
En esta categoría se integran dos comandos básicos: DIR y CHKDSK. Su función es proporcionar al usuario una idea de cuál es el contenido real del disquete sobre el que opera. En ciertas versiones del MS/DOS aparece un tercer comando de este tipo: CAT, con funciones muy similares a DIR.

DIR

Suministra una idea completa de los ficheros contenidos en el disco. Sin embargo, no da ningún tipo de indicación



Dentro del grupo de comandos adecuados para la «administración del disco» se encuentra DIR. Su cometido es proporcionar una lista de los ficheros residentes en el disco que se especifique.



La información entregada por la máquina en respuesta a la orden DIR incluye datos relativos a la identificación de los ficheros que residen en el disco afectado, a su tamaño y a la fecha y su hora de creación o última actualización.

acerca de la existencia de lo que se denominan «ficheros ocultos». Estos últimos contienen determinadas zonas del sistema operativo. El comando DIR puede adoptar distintos formatos:

DIR A:

Confecciona una lista con todos los ficheros residentes en la unidad de disco A, indicando, en cada caso, el nombre, extensión del nombre, longitud del fichero en bytes, así como la fecha en la que el fichero fue creado o actualizado por última vez.

DIR nombre de fichero

Suministra la misma información que en el caso anterior, sólo que restringida a un único fichero.

Normalmente suele asociarse al comando DIR algunos interruptores que permiten efectuar una pausa cada vez que se llena la pantalla («/P» o «—P ») o eliminar la información acerca del tamaño y la fecha de creación del fichero («/W» o «—W»).

CHKDSK

Las funciones de este comando son similares a las del comando DIR, sólo que más condensadas. Al invocarlo, revela los siguientes datos:

— Espacio libre en el disquete
— Espacio ocupado por ficheros
— Discrepancias en el empleo del espacio de almacenamiento.
— Número de ficheros en el disquete
— Número de ficheros ocultos (si los hay).

Una de las funciones de CHKDSK es la de reparar cualquier defecto lógico en el uso del espacio de almacenamiento. En el caso del MS/DOS versión 1.0, tal operación se realiza de forma automática.

Comandos para operación con ficheros

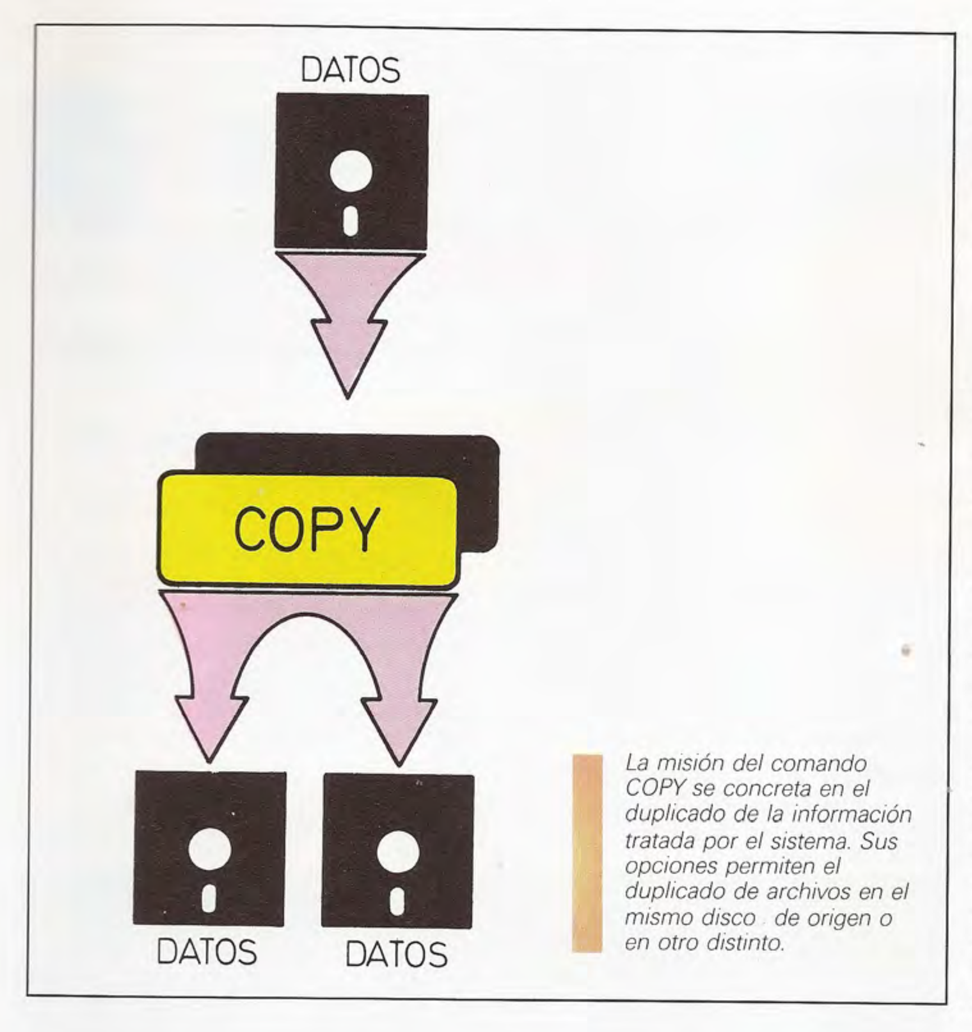
Estos comandos actúan sobre los ficheros modificando sus atributos externos, aunque sin modificar en su proceso de operación el contenido interno e intrínseco del fichero. Los comandos adscritos a este grupo son: COPY, DEL, REN y TYPE.

COPY

El comando COPY, tal como su propio



Los comandos para la administración del disco (DIR y CHKDSK) permiten al usuario obtener información inmediata y actualizada del contenido de los discos con los que opera.



La misión del comando COPY se concreta en el duplicado de la información tratada por el sistema. Sus opciones permiten el duplicado de archivos en el mismo disco de origen o en otro distinto.

nombre indica, se emplea para copiar ficheros. Sin embargo, introduce en esta función tan simple las suficientes variaciones como para servir a tres propósitos distintos:

— Duplicar información en distintos periféricos de almacenamiento masivo.

— Duplicar información bajo otro nombre.

— Combinar información procedente de varios ficheros en un único fichero.

Por lo que se refiere al primero de estos propósitos, el duplicado de información en distintos periféricos, bajo un mismo nombre, el comando COPY necesita que se le defina la información a copiar así como la unidad de disquete sobre la que se va a efectuar dicha copia. Cabe señalar que, en este caso, la unidad sobre la que se copia ha de ser distinta de la que contiene el disquete origen de la información. Ello se debe a que, en la versión DOS 1.0, no pueden existir dos ficheros con el mismo nombre sobre un mismo disco.

La forma en que se define el nombre del fichero que se va a copiar es variable, puesto que se admite el empleo de referencias ambiguas o «wildcards». Los ejemplos que siguen ilustran diversas posibilidades.

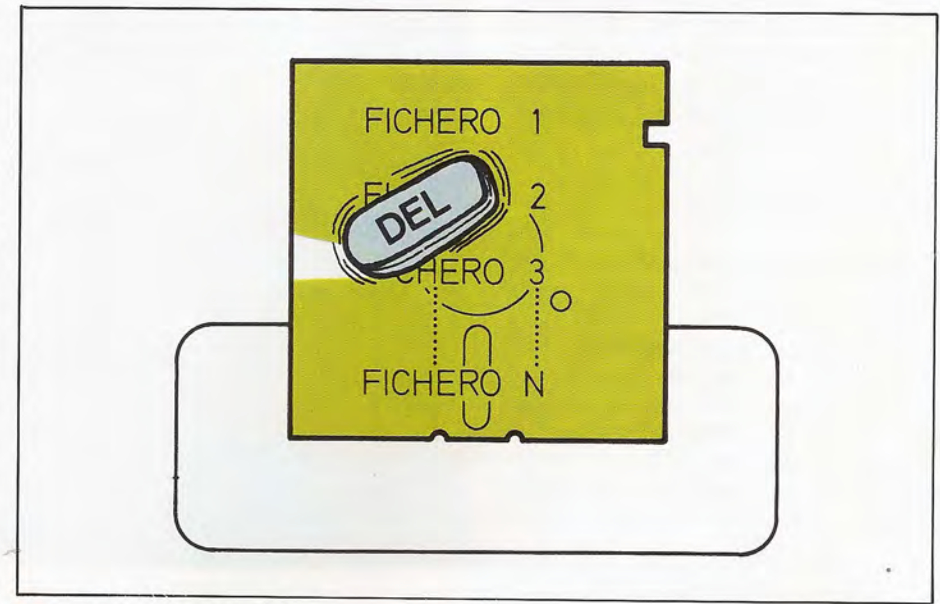
COPY A: DATOS B:

Este comando producirá una copia del fichero DATOS, contenido en el disquete de la unidad A sobre el disquete de la unidad B, con el mismo nombre.

COPY A: DATOS. *B:

En este caso, el empleo de las referencias ambiguas hará que sean copiados los ficheros que tengan como nombre propio «DATOS», sea cual fuere su tipo, y de acuerdo a las reglas indicadas en el caso anterior.

En lo relativo al segundo propósito del comando COPY, el duplicado de información con cambio de nombre, éste se consigue especificando el nombre del fichero que va a ser copiado, así como el nuevo nombre bajo el cual la información va a ser almacenada. De no especificar las unidades de disquete, el sistema operativo asumirá, por defecto, la unidad que en ese momento se consi-



El borrado de los ficheros es una tarea que cuenta con un comando especializado en el sistema operativo MS-DOS: ERASE.

dere como la propia del sistema. Por ejemplo:

COPY DATOS RESPALDO

Esta orden producirá una copia del fichero DATOS bajo el nuevo nombre de RESPALDO, residiendo ambos ficheros en el mismo disquete.

El tercer propósito del comando COPY, la agrupación del contenido de varios ficheros en un único fichero, depende en su formato de la versión del MS/DOS empleada. El formato genérico de esta orden suele adoptar el siguiente aspecto:

COPY A+B C

Tal y como está expresado, la ejecución del comando hará que los archivos A y B se copien uno tras otro, en un único fichero resultante denominado C.

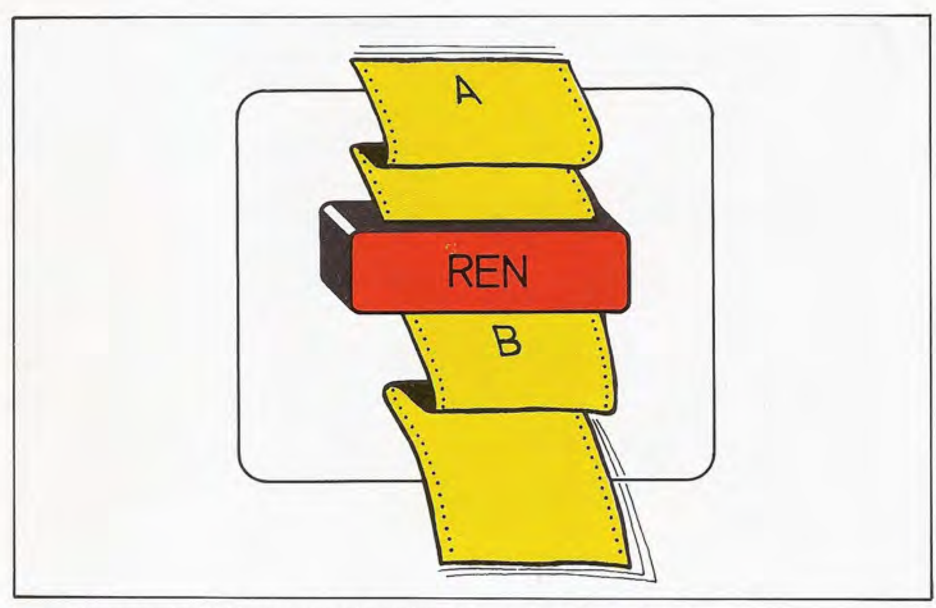
COPY es un comando potente y de gran utilidad que, sin embargo, presenta un peligro potencial. Este reside en el hecho de que si existe un fichero con el nombre otorgado a la copia, ésta se producirá sobre el mismo, sin proporcionar al usuario ningún aviso previo. Ello supone que datos valiosos pueden llegar a perderse. Por lo tanto, resulta conveniente emplear previamente el comando DIR, para garantizar que no se ha empleado con anterioridad el nombre del fichero que va a recibir la copia.

DEL

Este comando se presenta en ciertos equipos bajo el nombre de ERASE, e incluso en otros equipos se admite con ambas denominaciones. Su función es la de borrar o eliminar información. El procedimiento de operación exige definir el fichero que se desea eliminar. Para esta definición pueden emplearse también referencias ambiguas; no obstante, hay que señalar que dependiendo del grado de ambigüedad utilizado, así será el control que tendrá el usuario sobre la operación del borrado. En efecto, una orden del tipo:

DEL DATOS o DEL*. TEST

producirá la eliminación automática y



El comando REN gestiona el cambio de nombre de un fichero sin alterar la información contenida en el mismo.

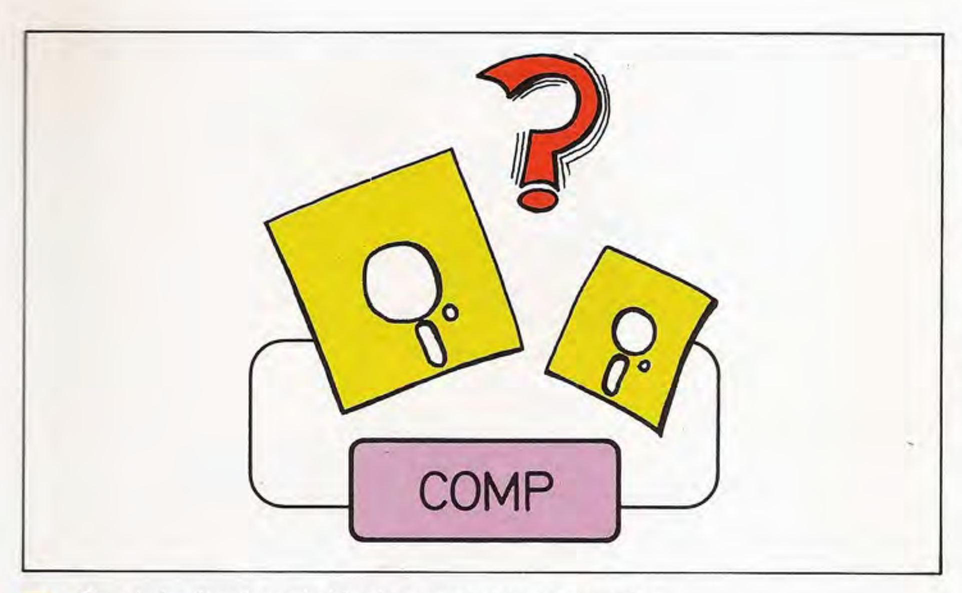
La comunicación ordenador-usuario

Cuando el usuario se sienta ante el terminal de un ordenador e inicia una tarea, da comienzo un proceso de intercambio de información con la máquina. El usuario introduce unos datos que el ordenador analiza en función de los dictados del programa. Como colofón, la máquina devuelve toda una serie de mensajes; información que el usuario considerará en orden a modificar los datos inicialmente introducidos para repetir de nuevo el proceso. Este diálogo proseguirá hasta que concluya la sesión de trabajo. Durante el periodo de actividad, el usuario necesita que la información procedente del ordenador le sea presentada con celeridad; independientemente del tiempo empleado por la CPU para el cálculo. Este hecho tiene un fuerte impacto en el diseño del ordenador, ya que determina la propia capacidad de los canales a través de los que dialogan el ordenador y el usuario. En efecto, si pensamos en un sistema de diseño gráfico, el usuario necesita que los diseños se visualicen inmediatamente en la pantalla. Como quiera que los trazados gráficos exigen un gran volumen de información y memoria, es preciso que los canales de comunicación sean de gran capacidad. Sin embargo, en tareas de programación durante las que el ordenador solicita esporádicamente información del usuario, la capacidad exigida a los canales es notablemente inferior.

Una forma de medir estas necesidades es recurriendo al cociente entre la máxima información transmitida por unidad de tiempo, y el volumen medio de información transmitida en el transcurso de una sesión de trabajo. Una tarea de programación tendrá un cociente bajo, mientras que un proceso gráfico lo tendrá muy alto.

Tipo de aplicación	Velocidad media de transmisión (bits/sg)	Velocidad punta de transmisión (bits/sg)
Cálculo corriente, empleando el lenguaje BASIC	8	100
Control de almacenes utilizando una base de datos	10	150
Entrada de datos a través de formatos de pantalla	15	1 500
Reserva de billetes por ordenador	10	2 000
Sistema gráfico en el que se utiliza un lápiz óptico para crear los diseños	200	200 000

Existen grandes diferencias entre la cantidad de información transmitida, por término medio, en el transcurso de una sesión de trabajo, y la información a transmitir en un breve período de tiempo para prevenir retrasos inaceptables por el usuario.



Otra de las funciones habituales en un sistema operativo y contemplada por el MS-DOS es la de comprobar ficheros de información; de ello se ocupa el comando COMP.

sin ningún tipo de solicitud de confirmación, en un caso del fichero DATOS y en el otro de todos los ficheros cuya extensión del nombre sea TEST. Sin embargo, un comando de la forma:

DEL *.*

solicitará la confirmación para cada uno de los ficheros a eliminar.

REN

El comando REN mantiene una estrecha relación con el comando COPY. Su función es la de cambiar el nombre de un fichero, siempre y cuando el nuevo nombre no sea ya utilizado por otro fichero. Su formato general es el siguiente:

REN NOMBREANTIGUO
NOMBRENUEVO

Al igual que otros comandos, REN también admite el empleo de referencias ambiguas, actuando en tal caso sobre varios ficheros.

REN *.* TEST*.PRUEBA

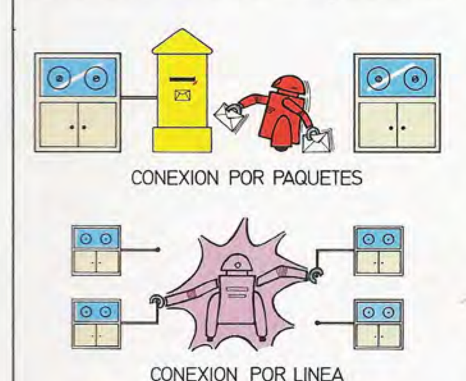
TYPE

La función de TYPE es comprobar el tipo de datos que contiene un fichero, proporcionando una visión del mismo e indicando la naturaleza de los caracteres contenidos (por ejemplo, ASCII).

La comunicación entre ordenadores distantes

En la actualidad, la tendencia primaria en el campo de la informática es la de brindar a cualquier usuario la posibilidad de acceder a los recursos de un ordenador, aunque éste no se encuentre físicamente próximo. De ahí nace el denominado proceso distribuido, por el cual, máquinas de distinta capacidad y potencia se interrelacionan entre sí, compartiendo datos, programas y recursos. Dependiendo de la distancia que medie entre los ordenadores asociados, se emplean diversos métodos de comunicación, tanto a nivel lógico como físico. Entre los métodos adecuados para la comunicación a larga distancia, los dos más comunes son los que se conocen bajo apelativos de «conexión por paquetes» y «conexión por líneas».

En el primer caso, la información a transmitir se divide en partes, denominadas paquetes, cada uno de los cuales lleva una etiqueta en la que se indica su destino. En la línea de comunicaciones existen ciertos elementos, denominados controladores, cuya función es leer las etiquetas y pasar el paquete a su destinatario, o al siguiente controlador más próximo. Cuando el paquete llega a su destino se decodifica y la información se almacena en el sistema. En el segundo método, el controlador se encarga de facilitar una comunicación directa entre los dos ordenadores que van a intercambiar información durante un periodo de tiempo extremadamente breve. En el transcurso de este intervalo de tiempo, se envía toda la información a transmitir de un sola vez. Este método tiene la desventaja de que depende drásticamente del número de líneas disponibles para habilitar el diálogo. De ahí que pueda darse el caso de que se intente la comunicación y no haya ninguna línea disponible para establecerla en ese momento.



Comandos de comparación

El MS/DOS ofrece dos comandos adecuados para efectuar comparaciones, bien entre ficheros o bien entre disquetes completos. Los comandos al efecto son COMP, el cual compara el contenido de dos ficheros entre sí, y DISKCOMP cuya función es la de comparar el contenido de dos disquetes.

COMP

Este comando en uno de los que presenta mayor variedad de denominaciones en el sistema operativo MS/DOS. Las más comunes, aparte de la ya señalada, son las de FC y FILE COMP.

La función que lleva a cabo es la de comparar entre sí el contenido de dos ficheros, comprobando si ambos coinciden con total exactitud. Esta comparación se efectúa sobre el contenido lógico del fichero, verificando su correspondencia, si bien no tiene en cuenta las posiciones respectivas de los ficheros en el disquete o disquetes en los que residen.

COMP puede presentarse en dos versiones, una «básica» y otra que cabe denominar «avanzada». En la versión «básica», el comando COMP únicamente indica si los dos ficheros sujetos a com-

paración difieren o no, pero no da ningún tipo de información adicional, con lo que el usuario no sabe cuál es el motivo de las diferencias. En la versión «avanzada», el comando COMP permite señalar una serie de características y hacer caso omiso de otras. En particular, cuando se están comparando ficheros que contienen texto, este comando permite ignorar si se ha escrito en mayúsculas o minúsculas para la comparación, o bien si existen espacios en blanco. Asimismo, permite tomar como origen el punto en el que ambos ficheros difieren, y a continuación iniciar un proceso de búsqueda hasta encontrar un punto en el que de nuevo vuelvan a coincidir de manera global.

Estas funciones es posible seleccionarlas por medio de interrupciones en el comando, lo cual es, precisamente, la forma de descubrir si la versión del MS/DOS utilizada es la «básica» que no dispone de ellos, o la «avanzada».

DISKCOMP

La misión de DISKCOMP es comparar si dos disquetes son exactamente iguales. El comando no opera en base a una comparación de tipo funcional, sino que también compara físicamente. Es por ello que aunque dos discos contengan exactamente la misma información, pero distribuida de forma distinta en cada uno de ellos, con lo cual sería funcionalmente compatibles y sustituibles, el comando DISKCOMP los señalará como distintos. Este comando sólo tiene interés si existe una necesidad imperiosa de comprobar si dos discos son totalmente iguales.

Comandos de operación con programas

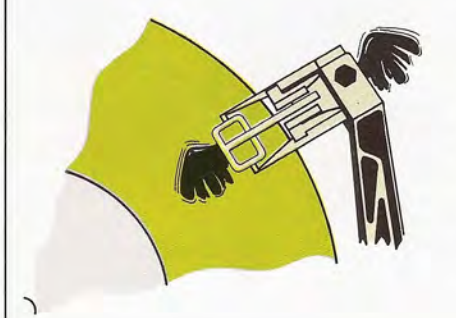
Los comandos que el MS/DOS destina al manejo de programas, a nivel de sistema operativo, pueden dividirse en dos grupos. Comandos destinados a la reunión de los diversos módulos de un programa y comandos destinados a la modificación de los formatos con que los programas se almacenan en un disco o disquete.



Cumpliendo el cometido propio de un sistema operativo, el MS-DOS pone al alcance del usuario la posibilidad de explotar con eficacia los recursos del ordenador.

Los discos rígidos

En los últimos años, los discos denominados «rígidos» han alcanzado una gran difusión como método para el almacenamiento permanente de información en los microordenadores. La velocidad de acceso y su elevada capacidad de almacenamiento —de 5 a 40 Mbytes por unidad— supone una sustancial mejora respecto a los discos flexibles. Hasta ahora, el tipo habitual de discos rígidos son de tecnología Winchester. Este apelativo procede del nombre en clave que empleó IBM, la firma creadora de esta tecnología, para denominar el proyecto. Los discos Winchester son semejantes en su constitución y funcionamiento a los discos de tecnologías anteriores. La mayor divergencia aparece en el método de operación de la cabeza grabadora de datos. El procedimiento, original y altamente eficaz, hace que los discos de esta tecnología resulten más fiables y duraderos. La clave estriba en el hecho de que la cabeza no establece contacto con la superficie del disco, sino que se encuentra separada del mismo a una distancia de medio micrómetro. Para conseguirlo, la cabeza está dotada de tres raíles paralelos. El raíl central es el que se emplea para la grabación y recuperación de los datos, mientras que los laterales gobiernan el flujo de aire. De esta forma, al girar el disco se produce una fuerza aerodinámica que separa la cabeza de la superficie magnética.



El conjunto se almacena en el interior de una caja hermética, en la que se introduce el aire filtrado, a presión. Este se encarga de eliminar las partículas de polvo que podrían interferir en el proceso de lectura y escritura de datos.



LOCATE y EXEFIX son dos comandos del MS-DOS cuya especialidad se concreta en la modificación del formato con el que se almacenan los programas.

- *Comandos de construcción de programas*

Básicamente son dos: LINK y LIB. El primero de ellos tiene la misión de agrupar el conjunto de elementos que constituyen un programa, como pueden ser el programa principal, las subrutinas que éste utiliza, así como las utilidades necesarias para la operación del mismo; entendiendo por utilidades al conjunto de programas y subrutinas estándar que llevan a cabo determinados trabajos de forma rutinaria. Esta tarea de unión la lleva a cabo el comando LINK, asociado a la operación con lenguajes no interpretados sino compilados.

El comando LIB es un comando menor cuyo cometido es el de mantener en condiciones adecuadas las diferentes librerías de utilidades del usuario, permitiendo la creación de librerías o la eliminación de partes de las mismas.

- *Comandos de conversión de formatos*

Cuando un programa se almacena de forma que quede dispuesto para un uso inmediato, lo hace bajo dos posibles formatos; estos son EXE y COM. El formato EXE es el empleado para aquellos programas que en su operación necesitan el pleno empleo de los recursos asociados al MS/DOS, mientras que el formato COM es el empleado con los programas que no exigen este tipo de ayuda; en consecuencia, este último hace un uso más rentable del espacio de almacenamiento en disco.

Dos comandos, LOCATE y EXEFIX, llevan a cabo la tarea de pasar de un formato a otro. LOCATE efectúa el paso de formato EXE a COM, mientras que EXEFIX opera en cambio contrario, de formato COM a EXE.



LINK permite agrupar en un sólo bloque a los distintos elementos de que consta un programa.

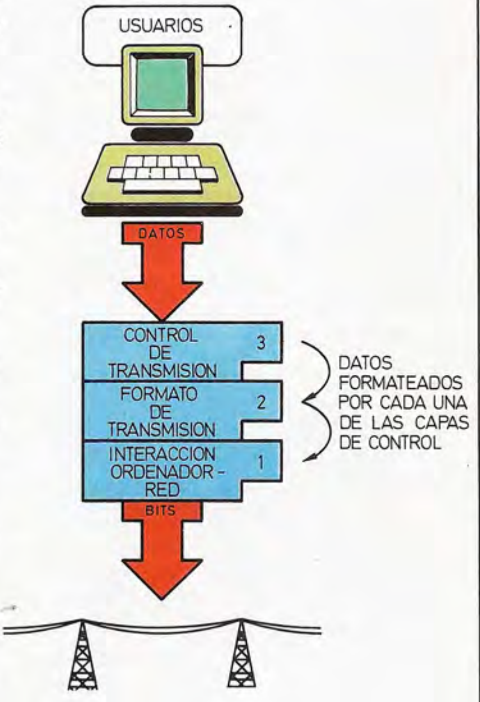


La comunicación: un eslabón crítico de la sociedad informatizada

La sociedad informatizada es una idea altamente atractiva y preconizada por muchos estudiosos. En ella la persona trabaja en muchos casos desde su propio hogar, por medio de un ordenador personal conectado a una red que lo comunica con su puesto de trabajo. A través de este canal, el usuario accede a la información del ordenador de su empresa o incluso a la almacenada en otros ordenadores. Ante el auge actual de la informática ésta es una realidad que no parece muy lejana. Sin embargo, hay un problema de base que sólo recientemente se está abordando: la comunicación entre ordenadores distintos, y distantes en su emplazamiento físico.

El proceso de comunicación entre los ordenadores cabe dividirlo en tres capas concéntricas, la primera de las cuales concierne a la relación física entre la máquina y el dispositivo de comunicación. Actualmente, la norma EIA RS/232-C o la recomendación CCIT V.24 brindan un estándar aceptable para este nivel de comunicación. La segunda capa en el proceso de comunicación está relacionada con la forma de envío de los datos a través de la línea física y con el modo de intercambio de los mismos. En este caso existe también un estándar, el procedimiento HDLC, establecido por la Asociación Internacional de Normas. No obstante, hay que señalar en este punto que cada fabricante tiene sus propias variantes que difieren ligeramente entre sí.

La tercera capa de la comunicación es el control de la transmisión. Aquí es donde se produce la ruptura, ya que cada fabricante emplea procedimientos de control que le son propios y normalmente incompatibles con los de otros fabricantes; tal es el caso de las arquitecturas SNA de IBM o DECNET de DIGITAL. Los dos niveles inferiores aparecen estandarizados en la mayoría de los casos, aunque bien es cierto que si la forma en que se envían los datos desde el nivel superior es particular del fabricante, poco se ha avanzado. Este es, en definitiva, uno de los puntos críticos en el proceso de construcción de la futura sociedad plenamente informatizada.



MS-DOS (2)

Otras versiones del MS-DOS: 2.0 y sucesivas



La aparición de discos rígidos con capacidades del orden de decenas de Megabytes ha tenido mucho que ver en la reorientación del desarrollo del sistema operativo MS/DOS. Esta es una incidencia ostensible a partir de la versión DOS 2.0. La disponibilidad de un gran volumen de datos residentes en un disco rígido tipo Winchester, ha hecho que las nuevas versiones del MS/DOS incluyan poderosas y flexibles herramientas para el control y el manejo de estos datos. Estas no eran contempladas en versiones anteriores, ya que al realizarse el almacenamiento principalmente en disquetes, no parecía razonable articular un complejo proceso de tratamiento para un volumen de datos relativamente pequeño.

Entre las novedades que aporta la versión 2.00 del MS/DOS en lo relativo al control del flujo de datos, destacan la organización de datos en directorios y subdirectorios, el direccionamiento de las entradas y salidas de un programa, la concatenación de programas empleando como ficheros de entrada de datos los ficheros de salida del programa ejecutado anteriormente y, finalmente, el cambio de consola del sistema. A continuación se describen individualmente cada uno de estos puntos.

La estructura arborescente de datos

El almacenamiento de datos en archivos residentes en disco (ya sean rígidos o flexibles) no está exenta de problemas. La localización de un determinado elemento de información puede que no sea tan clara como en un principio cabría pensar, debido a la cantidad de información almacenada y la ambigüedad de ésta, sobre todo en sistemas que disponen de un disco rígido de gran capacidad. Para resolver esta complicación, el MS/DOS versión 2.00 permite crear dentro de un mismo disco distintos directorios que pueden contener archivos o a su vez, nuevos directorios (también llamados subdirectorios). Así pues, ahora se puede crear una estructura lógica con la información y almacenar los datos que se refieren a un tema específico en su propio directorio. Ello contribuirá, sin lugar a dudas, a disponer de una estructura clara y ordenada de los datos.

Cualquier disco contiene un directorio maestro o raíz, creado a la hora de formatear el disco, del cual pueden derivar nuevos subdirectorios capaces de acoger ficheros en grupos completamente estancos. Ello da nacimiento a una estructura de información en forma de árbol.

La inclusión de un nuevo directorio se realiza mediante el comando MKDIR (MD en forma abreviada); su activación crea un directorio vacío. Una vez creado el directorio, éste se puede ir rellenando con archivos o subdirectorios cuyos nombres pueden ser iguales a los nombres ya existentes en otros directorios. Al respecto, hay que precisar que el nombre de cada fichero está determinado por su nombre y, además, por los nombres de los directorios que lo contienen.

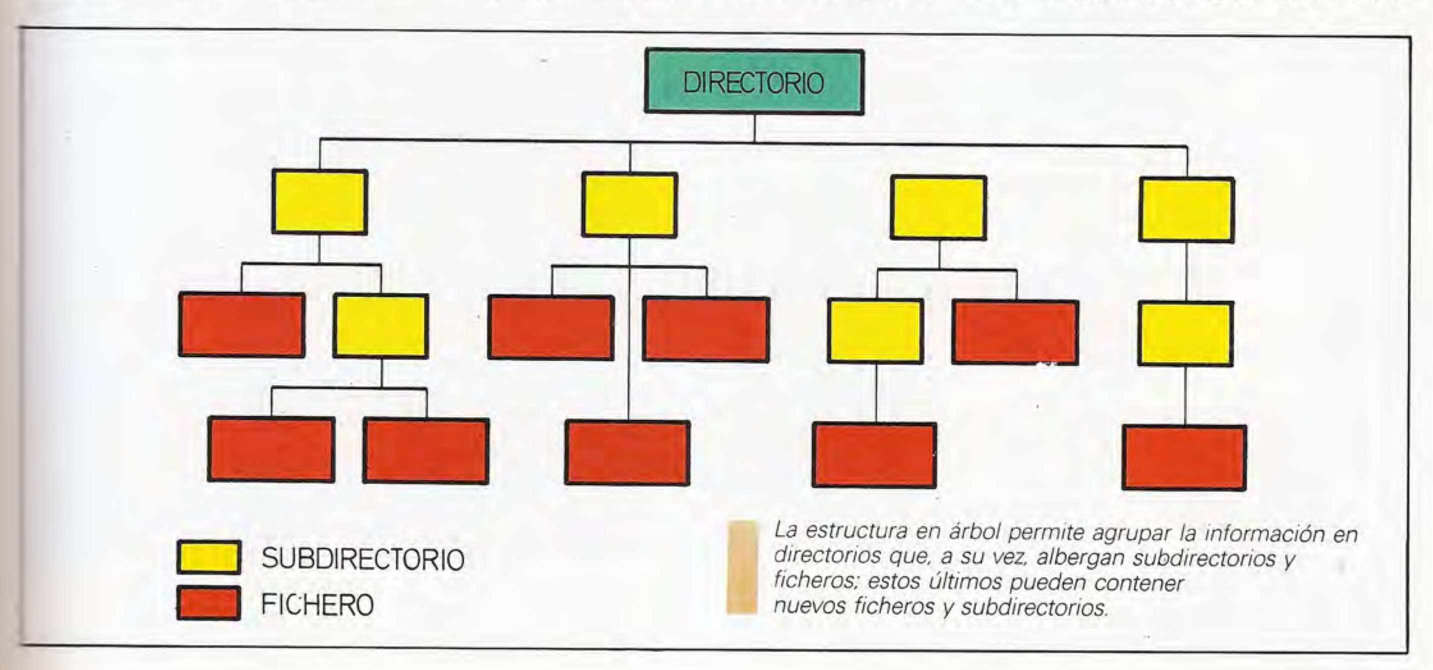
Dado que un disco puede incluir numerosos subdirectorios, organizados en una estructura de tipo árbol, hay que establecer un criterio para poder indicar en qué rama de la estructura se desea actuar. Para ello se define la noción de camino (path) como «la ruta que habría que seguir a través de los distintos directorios para llegar a la zona deseada».

Suponga que nos encontramos ante un disco con su directorio raíz que contiene, entre otros, un directorio llamado COMARCA; este último contiene a su vez otro llamado HOBBITON, y dentro de éste último hay un fichero de nombre BOLSON. Para describir el camino que conduce hasta el fichero BOLSON habrá que proceder como sigue:

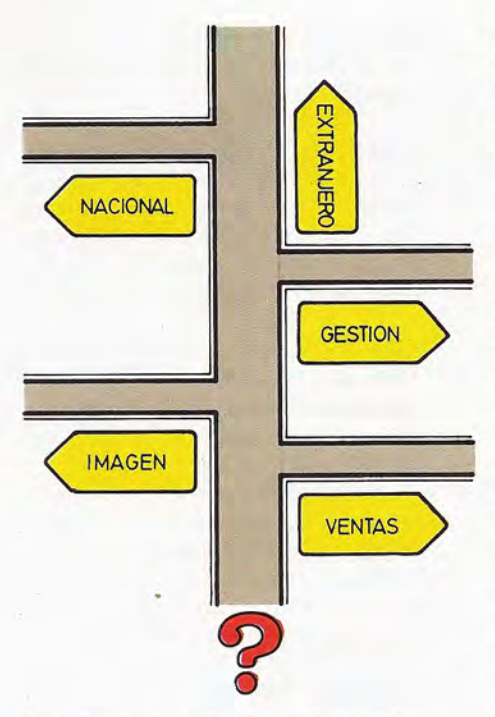
— Empezando en el directorio raíz, buscar el subdirectorio COMARCA.

— Localizar el subdirectorio de COMARCA cuyo nombre es HOBBITON.

— Por último, buscar dentro del subdirectorio HOBBITON el fichero BOLSON. La separación entre los distintos



La estructura en árbol permite agrupar la información en directorios que, a su vez, albergan subdirectorios y ficheros; estos últimos pueden contener nuevos ficheros y subdirectorios.



La tarea de localizar información resulta más racional si ésta queda integrada en una estructura arborescente.

nombres de los subdirectorios y los ficheros se hace mediante una barra inclinada o «slash» por lo que la representación del camino a seguir coincidirá con:

/COMARCA / ALTCAMPS / VALLS

La definición de los caminos puede ser absoluta si se parte del directorio raíz, o relativa si se comienza desde alguna rama del árbol. Un ejemplo de direccionamiento absoluto será el anteriormente descrito, ya que el directorio raíz se indica al empezar el camino con una barra inclinada. Por el contrario, un ejemplo de direccionamiento relativo sería:

ALTCAMP / BAG-END/ ALIO

El trabajo continuo con caminos puede resultar molesto si el trabajo se desarrolla en un subdirectorio determinado, ya que los nombres de los ficheros sobre los que se va a actuar tendrían que llevar siemper el prefijo indicativo del camino para llegar hasta ellos. Este inconveniente puede obviarse, ya que se puede indicar al MS/DOS cuál es el directorio sobre el que se desea actuar y éste nos posicionará sobre dicho directorio. De esta forma, una vez ubicados en el directorio elegido, ya es totalmente innecesario indicar el camino para llegar a él.

El comando que realiza la mencionada función es CHDIR o más abreviadamente, CD. Por ejemplo, para hacer que el subdirectorio «/COMARCA/ALTCAMP» sea el directorio sobre el cual se desea operar, bastará con escribir el comando:

CHDIR/ COMARCA/ ALTCAMP

Asimismo, también es posible cambiar de directorio si se dispone de varias unidades de disco y se desea pasar de un subdirectorio que está en un disco a otro subdirectorio localizado en otro disco. Para ello, es suficiente con referenciar el disco implicado antes de apelar al directorio. Eso es:

CHDIR A:/ MORDOR/ MONTE/ MONTEDESTINO/ ANILLO

Otros comandos relacionados con la estructura en árbol de la información son:

— RMDIR o RD: para borrar un directorio que previamente se ha vaciado de ficheros y subdirectorios.

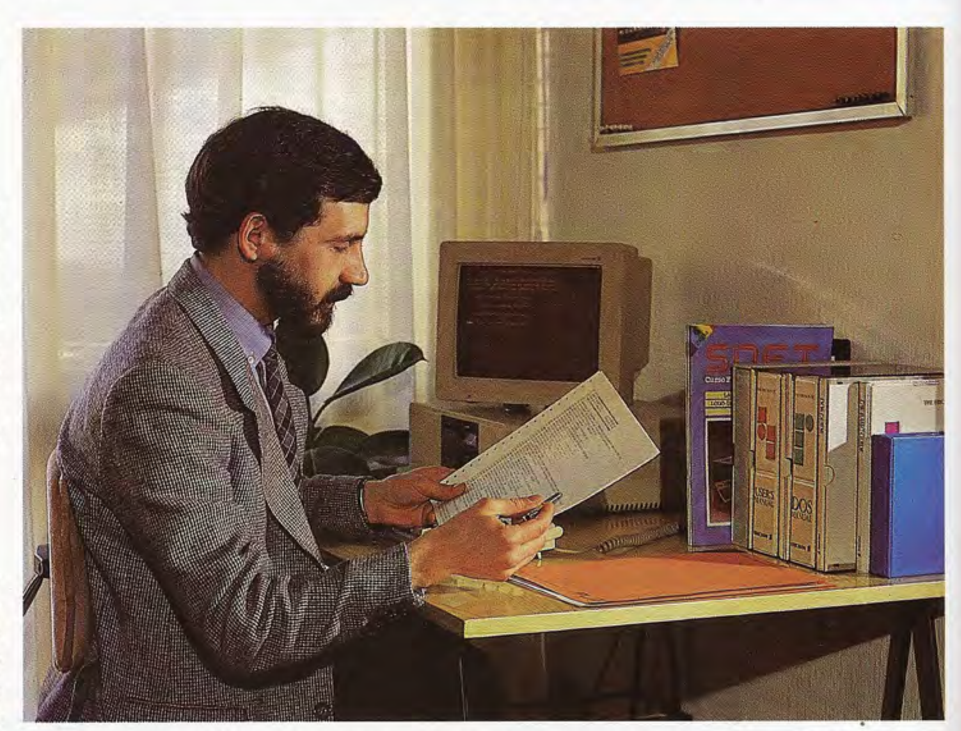
— TREE: muestra una lista de todas las ramas del árbol que cuelga del directorio raíz, para todas las unidades de disco conectadas.

— PWD: indica cuál es el directorio sobre el que está posicionado el sistema.

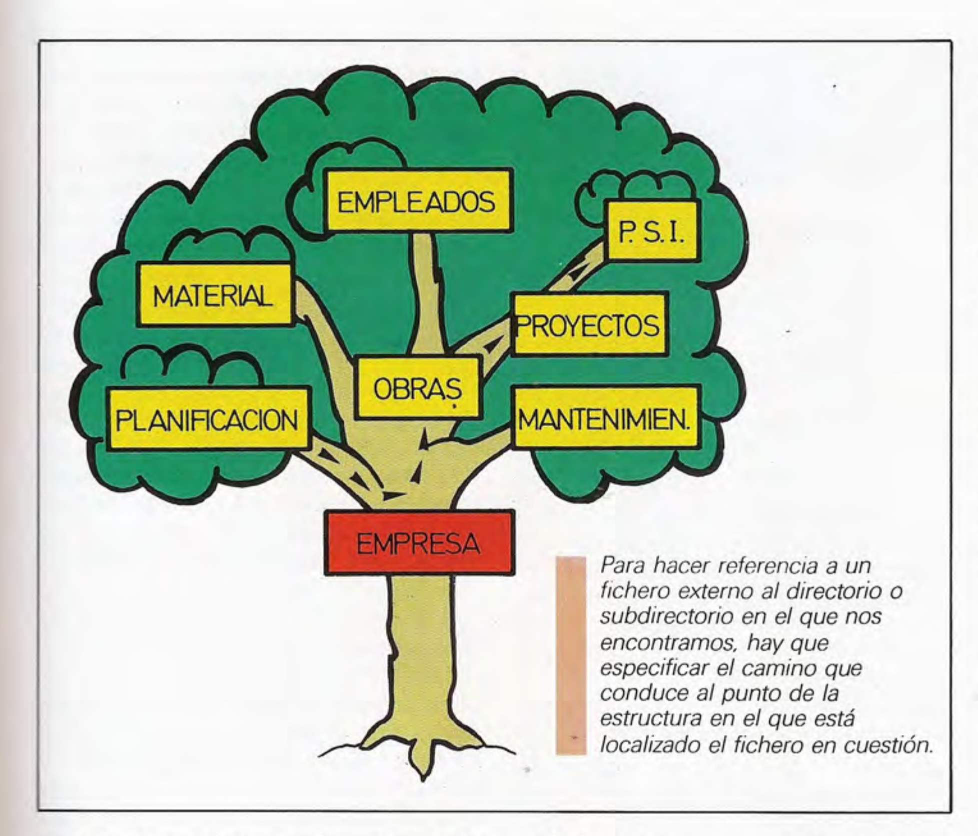
— PATH: define los caminos a través de los que han de buscarse los programas.

Otros modos de comunicación

Normalmente, cualquier programa exige la introducción de los datos necesarios para su ejecución a través del teclado y, a su vez, hace patentes los resultados visualizados en la pantalla asociada al sistema. Bajo la versión 2.00 del MS/DOS, la exposición exterior sigue siendo cierta, si bien tiende a ser más general. El programa lee información de una entrada estándar y plasma los resultados en una salida estándar, teniendo ambas la particularidad de



La versión 2.0 del MS-DOS brinda al usuario la posibilidad de organizar la información en estructuras lógicas arborescentes. Esto es una técnica de suma eficacia a la hora de organizar racionalmente grandes volúmenes de información.



Para hacer referencia a un fichero externo al directorio o subdirectorio en el que nos encontramos, hay que especificar el camino que conduce al punto de la estructura en el que está localizado el fichero en cuestión.

que no necesariamente tienen por qué coincidir con el teclado y la pantalla, sino que pueden asignarse otros dispositivos a estas entradas y salidas.

En general, el redireccionamiento de las entradas y salidas estándar se realiza con ficheros o dispositivos que memoricen la información el tiempo suficiente como para que ésta pueda ser revisada o actualizada en cualquier instante.

Suponga que nos encontramos ante un programa que proyecta los resultados de salida por pantalla y se desea que esta información quede archivada. Para lograrlo, será preciso corregir el programa para que los resultados queden ahora registrados en un fichero o se escriban a través de una impresora. Sin embargo, utilizando la posibilidad de redireccionar la salida, se puede conseguir que los resultados queden grabados en un fichero sin necesidad de alterar el programa. El método consiste, simplemente, en especificar el nombre del fichero en el que se desean grabar los resultados a continuación de nombre del programa, por ejemplo:

NOMINA> ENERO

El signo > se reserva para indicar a dónde se desea dirigir la salida. Por el contrario, el signo < se utiliza para indicar de dónde procede la entrada de datos.

Igualmente, se puede conseguir que un programa reciba los datos procedentes de un fichero en vez de recibirlos desde el teclado:

NOMINA>DATOS

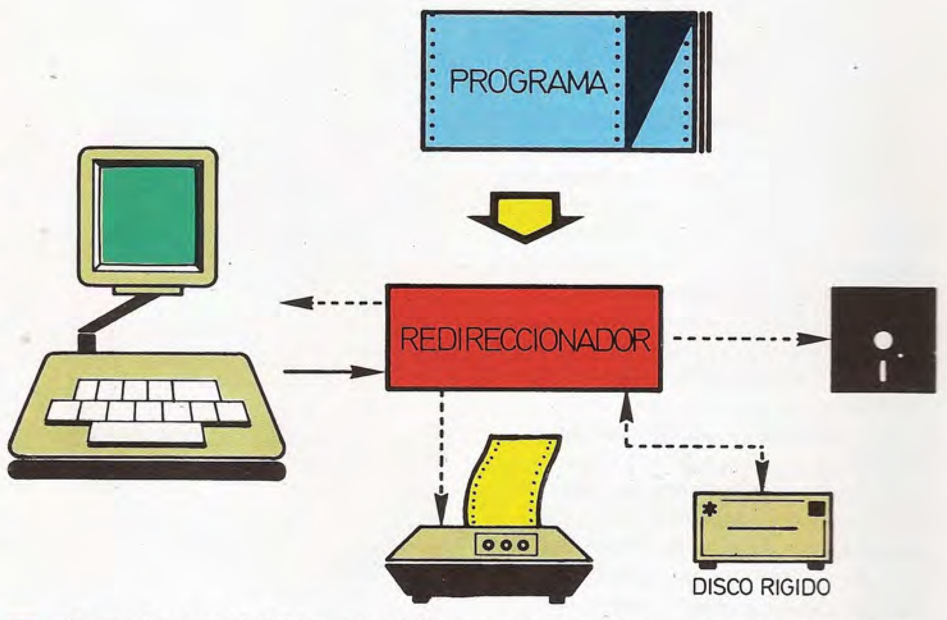
Y también es posible redireccionar la entrada y salida al mismo tiempo:

NOMINA>ENERO<DATOS

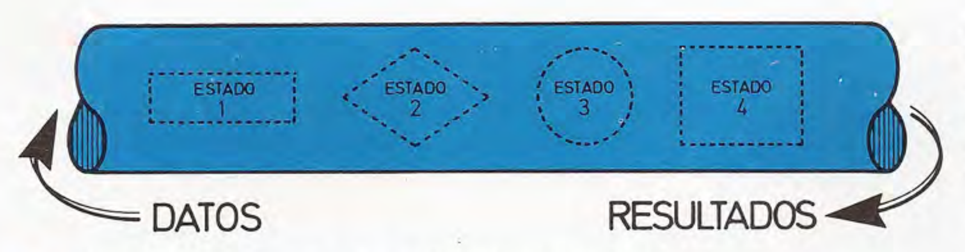
La razón principal para redireccionar la salida es almacenar la información resultante para una posterior consulta o revisión; mientras que el redireccionamiento de la entrada suele obedecer al hecho de evitar la necesidad de introducir los mismos datos si éstos tienen siempre un mismo formato.

Como construir un «programaducto»

Los programas manipulan datos en orden a resolver la tarea que se les ha encomendado. Y es probable, incluso, que un programa utilice como datos de entrada los datos de salida generados por un programa ejecutado con anterioridad. Con lo visto en el apartado anterior, se deduce fácilmente que el programa quedaría resuelto ejecutando el pri-



Por medio del MS-DOS versión 2.00 es posible modificar las entradas y salidas de un programa, sin que por ello haya que alterar el contenido del mismo.



Concatenando programas que actúen sobre los resultados de otros programas ejecutados anteriormente, puede construirse una cadena en la que los datos van pasando por sucesivos estados de transformación.

mer programa y redireccionando la salida a un fichero; a partir de ahí, sólo habrá que ejecutar a continuación el segundo programa, redireccionando su entrada al fichero creado como salida del primer programa. Esto es:

ESCRIBIR >TEXTO
CORREGIR<TEXTO

No obstante, el MS/DOS versión 2.00 ofrece una variante aún más sencilla para llevar a cabo este proceso, ya que puede encargarse de crear el fichero que sirve para intercambiar los datos y pasarlos de un programa a otro. Esto es lo que denomina cadena de programas o «pipeline» en la jerga del sistema operativo. Para crear un «pipeline» es suficiente con escribir los nombres de los programas separados por una barra vertical:

ESCRIBIR | CORREGIR

El formato de esta expresión puede hacer pensar que no se utiliza ningún fichero intermedio para pasar la información de un programa a otro. Ello no es cierto, ya que el sistema operativo se encarga de crear un fichero temporal en el cual el primer programa escribe sus resultados y del que el segundo programa lee sus datos; fichero que se borra automáticamente al finalizar la ejecución del segundo programa.

Hay una diferencia obvia entre los programas que ocupan la primera posición, la intermedia y la última dentro de esta cadena o «pipeline». El primer programa ha de ser necesariamente un generador de datos capaz de alimentar a los que vienen a continuación. Los programas emplazados en medio de la cadena deben consumir los datos generados por el programa anterior y, a su vez, tienen que generar otros datos para que sean utilizados por los programas que siguen. Y llegamos al extremo de la cadena. Sin lugar a dudas, el último programa debe ser un consumidor de datos nutrido por la información de salida de los programas que le preceden.

Dentro de una cadena de programas o «pipeline» cabe distinguir tres tipos de programas: generador de datos, modificador de datos y generador de resultados.

El cambio de consola

Los dos últimos métodos útiles para direccionar la entrada y salida de un programa, actúan mientras éste se encuentre en ejecución; una vez concluida la ejecución, puede recurrirse de nuevo a las vías tradicionales: el teclado y la pantalla. No obstante, ni se utiliza el comando CTTY, se puede conseguir que sea otro dispositivo quien haga las funciones de pantalla y teclado.

Este dispositivo debe ser capaz de actuar como un teclado y una pantalla a la vez; tal condicionante descarta a cualquier dispositivo para almacenamiento de información, como son los discos. Las condiciones exigidas las cumple, normalmente, un dispositivo de comunicaciones remotas, el cual puede pasar a constituir la consola del sistema:

CTTY AUX

De esta forma se puede conseguir un control remoto sobre el ordenador, quedando abierto el acceso al mismo desde un punto alejado al emplazamiento del ordenador, ya sea por vía telefónica o a través de cualquier otra línea de comunicación.

Los ficheros de comandos

Cualquier usuario de un ordenador se ha visto obligado a armarse de paciencia y aguardar expectante la conclusión de un proceso, para proseguir su actividad con la máquina de acuerdo al resultado obtenido. El MS/DOS resuelve estas ingratas esperas, a la vez que pone a disposición del usuario una herramienta que le permite «personalizar» el sistema operativo. La clave se encuentra en el uso de ficheros cuyo contenido

son comandos, expresados con el mismo formato que el que tienen cuando son introducidos a través del teclado. Dichos ficheros reciben el nombre de *ficheros de bloques de comandos* o «batch files».

Los ficheros de bloques de comandos constituyen una de las características más relevantes del sistema operativo MS/DOS. Estos ficheros, deben incluir obligatoriamente como nombre de extensión el sufijo «BAT», para que sean reconocidos como ficheros de comandos por el sistema operativo. Su contenido —un comando por cada registro— puede ser creado mediante un editor de textos, como el EDLIN, o con el auxilio de un procesador de textos, siempre y cuando dicho procesador produzca ficheros de texto ASCII sin ningún carácter de control especial.

La ejecución de un fichero de bloques de comandos se produce al introducir por el teclado el nombre del referido fichero; no es necesario en este caso especificar el nombre de la extensión. En consecuencia, el sistema operativo buscará primero entre los comandos internos, luego entre los externos, a continuación entre los ficheros cuya extensión sea COM y EXE y, finalmente, si el nombre especificado no se refiere a ninguno de los tipos anteriores, buscará entre los ficheros cuya extensión sea BAT. Una vez localizado el fichero, da comienzo la ejecución del comando o programa indicado en el primer registro del mismo. Tras procesarlo, se pasa a la ejecución del próximo comando, y así sucesivamente hasta alcanzar el final del fichero. Hay que tener en cuenta que el MS/DOS lleva la cuenta de los comandos ejecutados y siempre conoce cuál es el próximo comando a ejecutar.

Aplicaciones

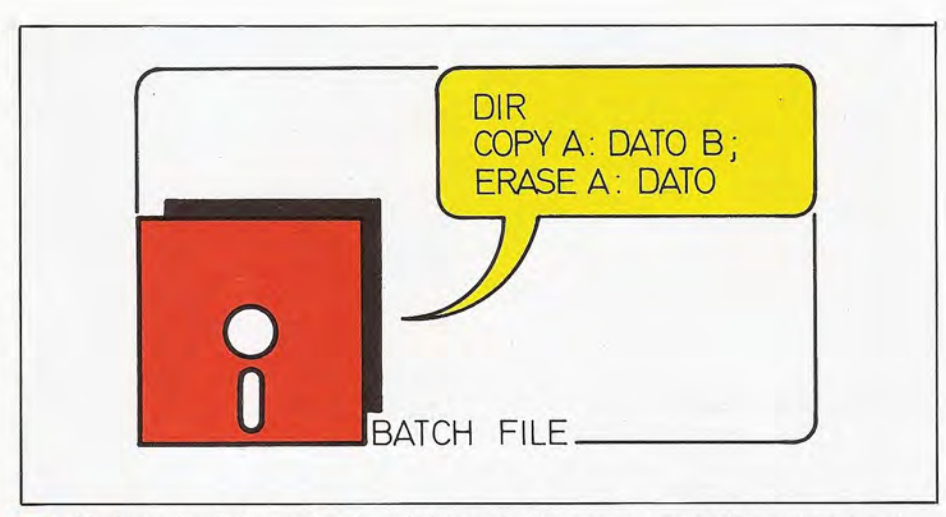
El uso de los ficheros de bloques de comandos permite al usuario crear un entorno de comandos a su medida. De esta forma, las operaciones más frecuentes o las que sigan siempre una misma metodología, son ejecutadas con el mínimo esfuerzo.

Una de las aplicaciones más evidentes, aunque no por ello menos interesante, es el cambio de nombre de los comandos, ya sea para adaptarlos al gusto del usuario, o bien para construir abreviaturas y nombres nemotécnicos que ayuden a simplificar su empleo.

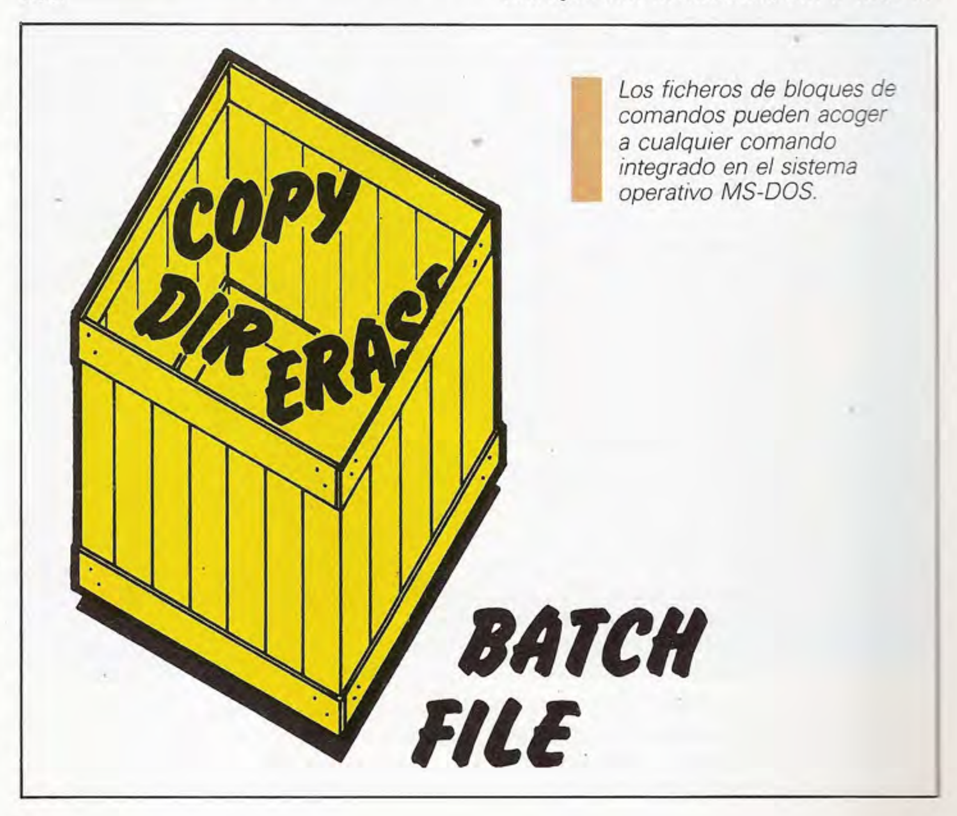
Por ejemplo, se puede crear un archivo de nombre DISCO.BAT, cuyo contenido coincida con los siguientes comandos:

DIR (Lista los ficheros del disco)
CHKDSK (Informa del estado del disco)

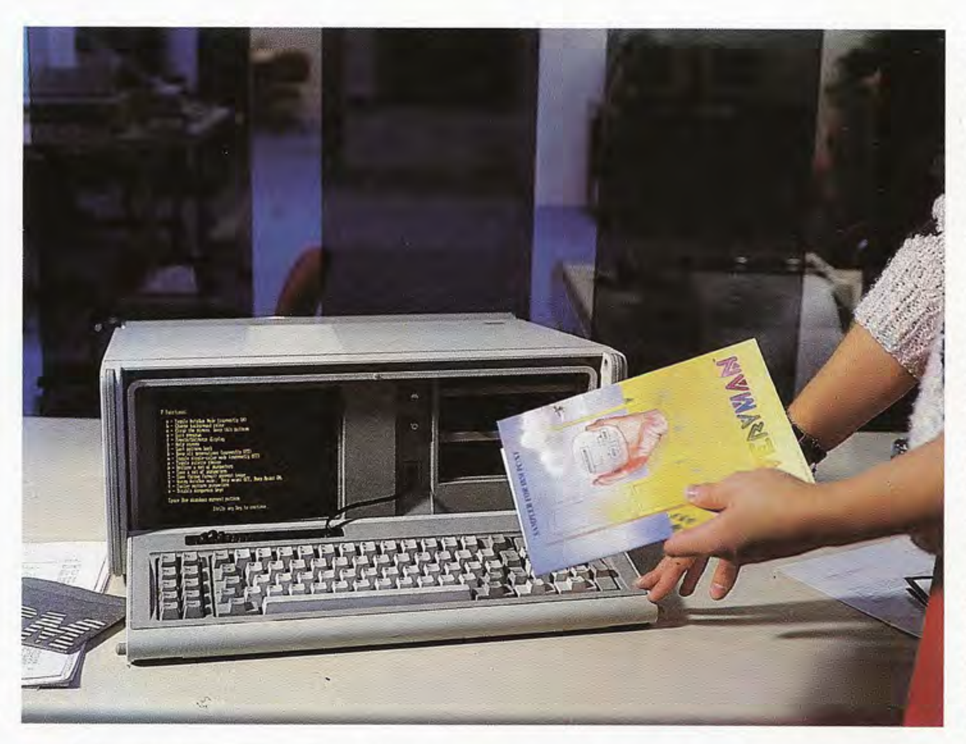
Así pues, cuando se teclee la palabra DISCO y se pulse la tecla ENTER, el resultado será una información completa del contenido y estado del disco. Las ventajas son indudables; sin ir más le-



Mediante los ficheros «batch» o ficheros de bloques de comandos, se instruye al ordenador para que ejecute, encadenadamente y de forma automática, una serie de comandos.



Los ficheros de bloques de comandos pueden acoger a cualquier comando integrado en el sistema operativo MS-DOS.



El uso de los denominados «ficheros de bloques de comandos» libera la constante atención del usuario sobre el ordenador. Un fichero de esta categoría puede contener toda una secuencia de comandos, cuya ejecución encadenada se producirá de forma automática con sólo llamar al fichero en cuestión.

jos, no hace falta recordar los nombres y la formulación de la serie de comandos necesarios para acometer una determinada tarea.

El uso de ficheros de bloques de comandos que encierren llamadas a otros ficheros de ese mismo tipo, exige adoptar ciertas precauciones. Hay que recordar que los ficheros de bloques de comandos pueden ir encadenados, pero nunca anidados. En otras palabras: suponga que un fichero de bloques de comandos, A, tiene en uno de sus registros el nombre de otro fichero de bloques de comandos, B; al terminar la ejecución del fichero B, el control no volverá al fichero A, sino que se interrumpirá la ejecución de dicho fichero en el registro siguiente al que hacía referencia al fichero B.

Las aplicaciones que requieren una serie de pasos englobados, dentro de una única unidad funcional, constituyen ejemplos típicos en los que resulta adecuado el uso de ficheros de bloques de comandos. En ellas, el fichero de bloques de comandos resumirá los diversos pasos a realizar, ordenados secuencialmente, de tal forma que cuando termine uno de ellos empezará la ejecución del siguiente. Como ejemplo, cabe citar la confección y puesta a punto de un programa en PASCAL, el cual requiere los siguientes pasos: a) creación del fichero fuente en PASCAL, b) compilación, c) edición de enlaces y d) ejecución: cada uno de estos paso estará en correspondencia con un registro del fichero de bloques de comandos.

Dada la gran variedad de casos tratables y las distintas necesidades de cada usuario, hay que concluir resaltando la flexibilidad seguridad y velocidad que aporta este método, dejando el campo abierto a la iniciativa del usuario.

Uso de parámetros

El uso práctico de los ficheros de bloques de comandos se flexibiliza en gran medida con el empleo de parámetros asociados. Estos permitirán que un mismo fichero de bloques de comandos sirva para distintas tareas, según sea el parámetro indicado. Los parámetros se colocan a continuación del nombre del fichero y sustituyen dentro de éste a los símbolos % 1, %2, %3... de acuerdo al lugar que ocupan los diversos parámetros en la línea de comando. Por ejemplo, si se desea construir un comando que copie todos los ficheros de un disco a otro, permitiendo la definición de los discos de origen y destino, puede crearse un fichero de bloques de comandos de nombre TRANS.BAT. Su cometido coincidirá con:

COPY1:*.*2:

Para activar su funcionamiento será

Con el comando PAUSE se consigue detener la ejecución del fichero de bloques de comandos, dando oportunidad al usuario para decidir el instante en que debe reanudarse el proceso.

necesario incluir dos parámetros junto al nombre. Así, por ejemplo, TRANS A B particularizará el contenido del fichero de comandos de la siguiente forma:

COPY A:*.*B:

Ordenando la copia de todos los ficheros del disco A en el B.

La presencia de los parámetros permite elegir a voluntad los discos afectados, con la única limitación de que el número de parámetros no puede ser superior a nueve.

Comandos para ficheros de bloques de comandos

El contenido de un fichero de bloques de comandos no se limita exclusivamente a comandos del sistema, sino que también admite comandos que gobiernen la ejecución de estos ficheros Las revisiones del DOS anteriores a los 2.00 sólo admiten los comandos REM y PAUSE; no obstante, a partir de la revisión 2.00 son aceptados todos los comandos que se describen a continuación :

• REM

No tiene ningún efecto específico. Se utiliza únicamente para introducir comentarios que ayuden a interpretar la función de los comandos.

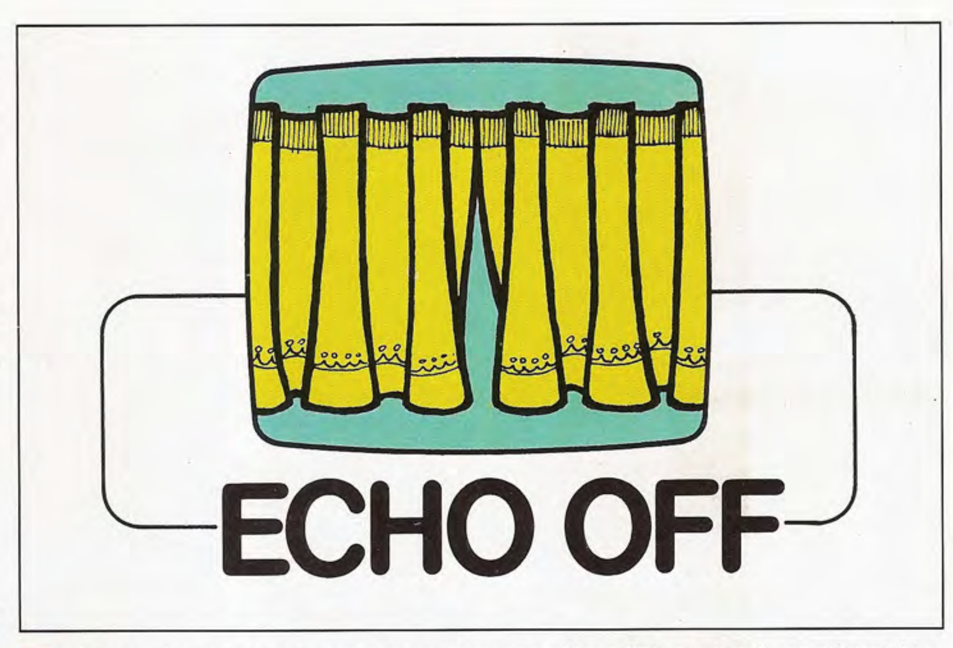
• PAUSE

Detiene la ejecución de un fichero de comandos; ésta no se reanuda hasta que se accione alguna tecla. La pausa puede ser de utilidad, por ejemplo, cuando está saliendo una gran cantidad de información por la pantalla, y no hay tiempo material para examinarla; su presencia detendrá el flujo de información temporalmente.

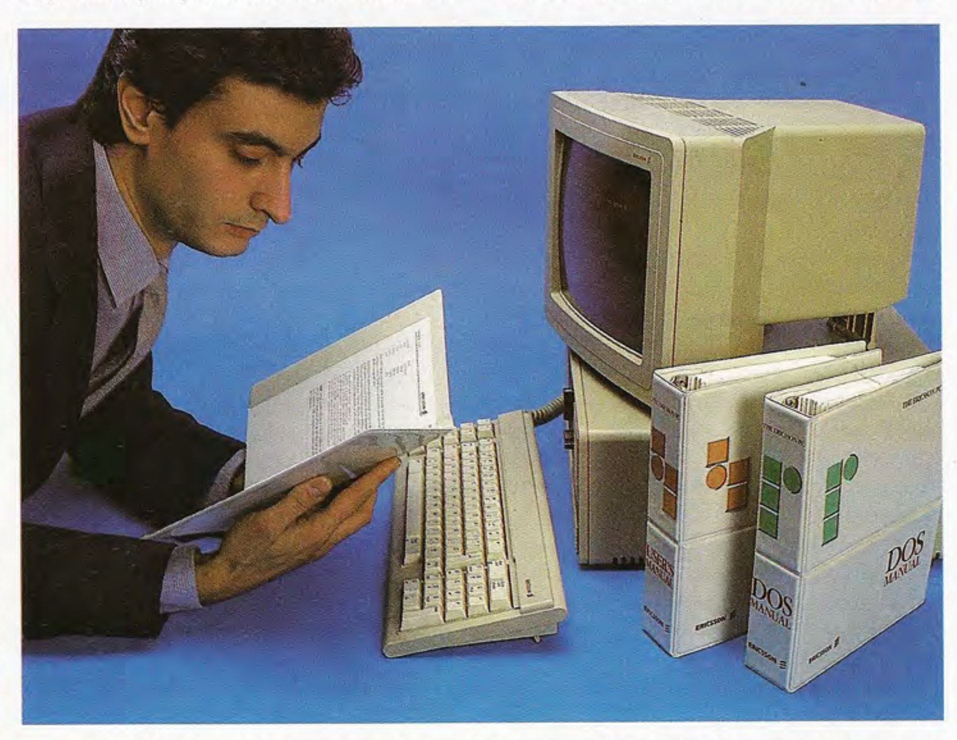
A menudo no se desea que la ejecución de un programa sea enteramente automática, por motivos de seguridad; algún acontecimiento imprevisto puede llevar a resultados catastróficos. Aquí también es de gran utilidad la inclusión de una pausa en los lugares críticos para poder controlar si se desea o no la realización del paso siguiente.

• ECHO

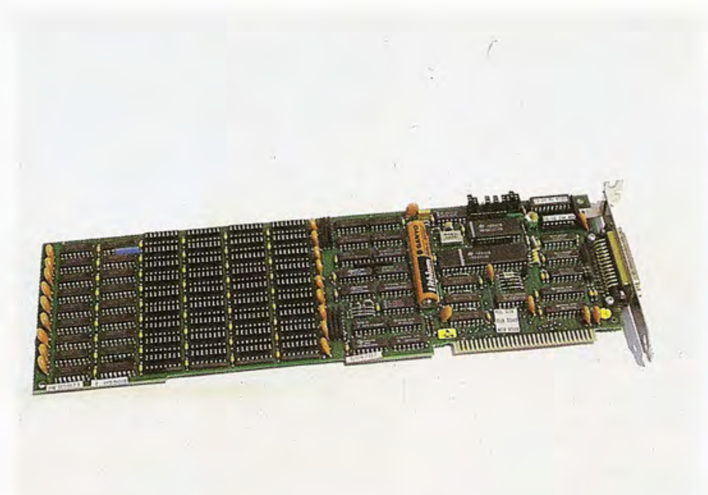
Este comando tiene dos estados: ON y OFF, activado y desactivado. Cuando está activo, el efecto que produce es la impresión por pantalla de los comandos que van siendo ejecutados, en caso contrario, no produce efectos visibles. La elección de uno u otro estado depende

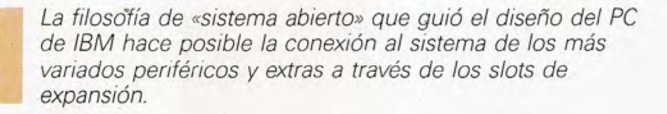


La visualización por pantalla de los comandos de un fichero «batch» es controlable por medio del comando ECHO. Los parámetros ON y OFF ordenan su activación o desactivación.



La posibilidad de trabajar con ficheros de bloques de comandos, permite al usuario crear un repertorio de órdenes evolucionadas a su medida. De esta forma, las operaciones más frecuentes o ajustadas a una misma metodología serán ejecutadas con el mínimo esfuerzo.





La filosofía de «sistema abierto» que guió el diseño del PC de IBM hace posible la conexión al sistema de los más variados periféricos y extras a través de los slots de expansión.



Las necesidades de cómputo para realizar trabajos en Inteligencia Artificial son satisfechas en gran medida por la potencia de los PCs y de su sistema operativo, el MS-DOS.

del objetivo que se persiga. En el caso de que se esté desarrollando una aplicación, interesará ver los comandos ejecutados; ello facilitará la depuración de los programas. En otras ocasiones puede resultar molesto o de mal efecto el hecho de que aparezcan los comandos ejecutados.

Un uso especial de este comando se produce al emplearlo para imprimir mensajes en la pantalla con el formato.

ECHO mensaje

El mensaje siempre aparece aunque el estado de ECHO sea OFF.

- SHIFT

Interviene en los casos en los que se manejan parámetros. Su efecto es lograr que todos los parámetros se desplacen un lugar hacia la izquierda, desapareciendo el primero de ellos. Esto es: el símbolo %1, toma el valor que tenía %2, %2 toma el valor de %3 y así sucesivamente.

- GOTO etiqueta

El efecto de este comando es variar el flujo secuencial de los comandos, dando un salto dentro del archivo de bloques de comandos hacia el lugar indicado por la etiqueta. La etiqueta se distingue por estar precedida por el signo dos puntos. Suele asociarse a sentencias condicionantes. Por ejemplo:

```
:OTRO
ERASE A: %1
SHIFT
GOTO OTRO
```

Su efecto será borrar todos los ficheros indicados en los parámetros.

- IF

Este comando comprueba una condición lógica; si es cierta ejecuta el comando situado a continuación de IF y en caso contrario lo ignora. El formato es:

IF Condición Comando

Algunas de las condiciones lógicas que se pueden comprobar son la existencia de un fichero, la ocurrencia de un error, la comparación entre dos cadenas de caracteres...

- FOR

Se emplea para ordenar la ejecución repetitiva de un determinado comando mediante una variable simbólica que va precedida por dos signos de «porcentaje» (%%). Esta variable adopta un valor de una lista de posibles valores.El formato es el siguiente:

FOR%% Nombre In (lista de valores)
DO VOL %% Nombre:

Por ejemplo:



Los paquetes integrados de aplicaciones han conocido un importante esplendor con la llegada del sistema operativo MS-DOS, cuyas prestaciones permiten la ejecución del citado tipo de software.



Los ordenadores portátiles y transportables permiten disponer de toda la potencia proporcionada por el MS-DOS en los momentos y lugares más variopintos.

FOR %% Nombre IN (A,B,C)
DO VOL %% Nombre:

alude a la etiqueta del volumen de los discos A, B y C.

La versión 3.00

Hasta el momento nos hemos detenido especialmente en la descripción de los comandos y características de la versión 2.00 del sistema operativo MS-DOS. Lo cierto es que la evolución que ha seguido este sistema ha traido como consecuencia la aparición de un gran número de revisiones del mismo, las cuales, a partir de la 2.00 apenas se diferencian si no es más que por la aparición de un puñado de nuevos comandos y alguna que otra modificación de los ya existentes. Como podrá comprobar el lector, el salto de gigante se dio con la aparición de la versión 2.00, en la que por primera vez nos encontramos con estructuras arborescentes de directorios y mecanismos de redireccionamiento de la entrada/salida. Se puede afirmar que la versión 1.00 apenas si era una revisión del viejo CP/M, mientras que a partir de ella surge un sistema operativo con características bien diferenciadas.

Las revisiones más populares que siguieron a las 2.00 fueron la 2.20 la 3.00 y la 3.10. Como ya se ha comentado, las diferencias entre ellas se centran en la aparición de algún que otro comando o en la modificación de la sintaxis o forma de actuar de otros. En las líneas que siguen profundizaremos algo más en la versión 3.00 del MS-DOS, que es la que por el momento conjuga con mayor éxito la existencia de ciertas novedades con relación a lo dicho hasta el momento y la disponibilidad para los usuarios.

Empezaremos comentando aquellos comandos existentes en anteriores versiones que han sufrido ciertas modificaciones en su forma de actuar.

El comando FORMAT, cuya misión es la inicialización del material magnético de los discos flexibles para que los datos sean «colocados» sobre tal soporte de una forma ordenada, ve aparecer como nueva opción el parámetro /4. Esto permite soslayar el problema asociado al formateado de un disco sobre equipos como los AT o compatibles, los cuales disponen de unidades de disco flexible de 1,2 Mbytes de capacidad. Si corriendo el MS-DOS sobre un AT se formatea un disco flexible, la información que posteriormente se grabe sobre él será absolutamente irreconocible para un PC o XT, en los que las unidades de disco consideran que los disquetes son de 360 Kbytes. Esto se debe al hecho de que el disco ha sido formateado a 1,2 Mbytes. Si se desea poder utilizar un disco grabado en un AT sobre los equipos menores de la gama de IBM, es necesario utilizar el parámetro/4 en la orden FORMAT.

Los comandos BACKUP y RESTORE han sido ampliados en el sentido de que



El respaldo proporcionado por IBM al sistema operativo MS-DOS se ha traducido en la ingente cantidad de software disponible para los usuarios de este sistema.

ahora es posible realizar todo tipo de transferencia entre unidades de disco, tanto flexibles como rígidos. En esta versión se permiten las siguientes combinaciones de medios magnéticos involucrados en la operación de estos comandos: disco rígido a disco flexible, disco rígido a disco rígido, disco flexible a disco rígido y disco flexible a disco flexible.

DISKCOMP y DISKCOPY, cuyas respectivas funciones consisten en realizar una comparación de dos discos sector a sector y la copia en iguales condiciones de dos discos, han sido ampliados al admitir como discos fuente y destino los que se encuentren en el interior de drivers de alta densidad, es decir, con capacidad para 1,2 Mbytes.

En una deferencia hacia usuarios no norteamericanos, el comando DATE, cuya misión es la fijación de la fecha del sistema, permite introducir la hora en cualquiera de los siguientes formatos: dd-mm-aa, mm-dd-aa o aa-mm-dd. La forma en la que interpretará la fecha tecleada por el usuario dependerá del valor de la variable del sistema COUNTRY, el cual se modifica en el fichero CONFIG.SYS. En concreto, el valor que le ha sido asignado a España es el 034.

Por último, el comando GRAPHICS, que instala en memoria el driver encargado de realizar los volcados por impresora del contenido de una pantalla gráfica, puede en la presente versión realizar su tarea sobre un número mucho más elevado de impresoras que en las versiones anteriores.

Hasta aquí llegan las modificaciones de comandos ya existentes. A partir de este momento se comentarán los nue-



El teclado sigue siendo el principal medio de introducción de datos en un sistema operativo tan potente como el MS-DOS, si bien otros medios alternativos como el ratón o la tableta gráfica están empujando con fuerza, sostenidos por las características de este sistema.

vos comandos característicos de la versión 3.00

En primer lugar nos encontramos con el comando ATTRIB, que se encarga de modificar el atributo de sólo lectura de un fichero. Cuando este atributo está fijado, el fichero afectado no podrá ser borrado o sobreescrito. En las anteriores versiones proteger un fichero de esta manera resultaba imposible si no se poseían ciertos conocimientos de la estructura más profunda de los ficheros del MS-DOS. Suponiendo la existencia de un fichero VITAL.DAT, al lanzar el comando:

A> ATTRIB + R VITAL.DAT

el citado fichero pasará a ser sólo de lectura, protegiéndolo de posibles errores. Si ahora se lanzara un:

A>ATTRIB VITAL.DAT

En la siguiente línea de la pantalla aparecería algo parecido a «R A:ÑVITAL.DAT», donde la «R» que aparece en primer lugar indica la circunstancia de que el fichero se encuentra en modo de sólo lectura. Para devolver a nuestro fichero la condición inicial de lectura-escritura, basta lanzar un:

A>ATTRIB -R VITAL.DAT

con lo que al lanzar de nuevo un «ATTRIB VITAL.DAT» el resultado sería básicamente el mismo pero sin aparecer en esta ocasión una «R» en el extremo izquierdo de la pantalla. Como se puede comprobar, la utilización del comando ATTRIB es algo tediosa. En la actualidad existen un buen número de pequeños programas de aplicación que son capaces de cumplir las funciones de este comando y muchas otras más de una manera bastante más elegante.

Otro comando que aparece por primera vez con la versión. 3.00 es el LABEL, que se encarga de dar una etiqueta a un disco flexible o rígido. Una etiqueta son una serie de caracteres que aparecen cada vez que se realiza un directorio en el disco en cuestión y que permiten identificarlo en el caso de que el adhesivo de la cubierta haya desaparecido. En general es una buena práctica el mantener todos los discos con su etiqueta correspondiente, ya que de esta forma se puede identificar su contenido más fácilmente. Su sintaxis es de las más simples, pues basta teclear LABEL seguido de la unidad cuya etiqueta se desea modificar. Acto seguido aparecerá un mensaje en pantalla pidiendo la introducción de los caracteres (hasta un máximo de once) que se desea formen la etiqueta.

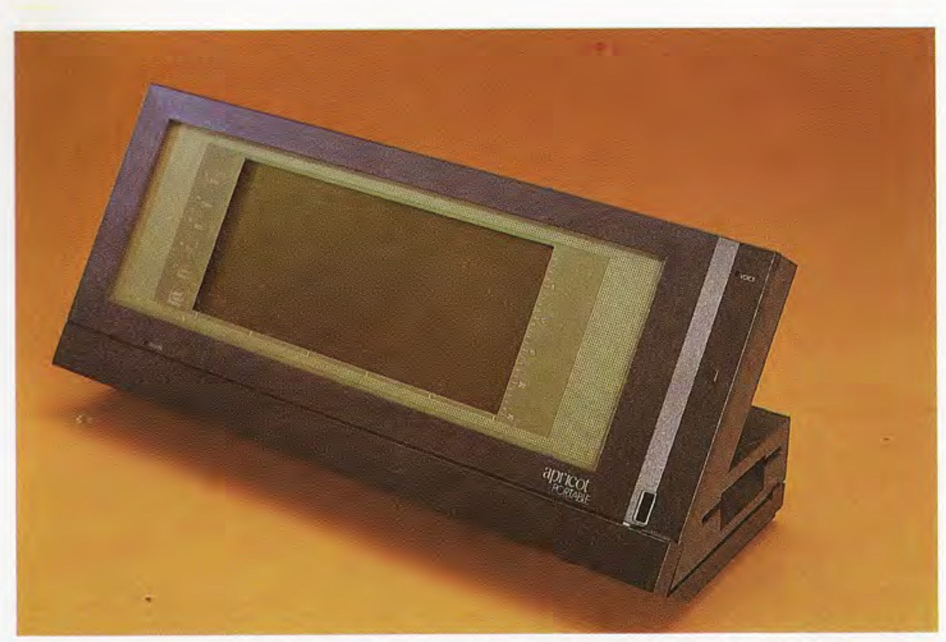


Cualquier sistema operativo, y por supuesto el MS-DOS, es un trabajo de ingeniería de software en el que se ven involucrados enormes sumas de dinero y de esfuerzo humano, medidas en horas de programación.

El comando SELECT se ha de aplicar en los primeros instantes de la recepción del equipo. Su misión consiste en realizar una copia del disco original del sistema y crear en él dos ficheros, un CONFIG.SYS que contiene como único texto «COUNTRY = xxx» y un AUTOEXEC.BAT con el único comando «KEYByy», en donde «xxx» significa el código de país que determina la presentación y lectura de la fecha del sistema por medio del comando DATE, e «yy» son dos letras que están relacionadas con el idioma al que se quiere configurar el teclado. En el caso de España, nada más recibir el equipo se deberá lanzar el comando SELECT 034 SP con el disco original de sistema en la unidad A: y un disco virgen en la B:. Al finalizar la operación, la segunda unidad de discos contendrá un disquete con el que se puede arrancar el sistema, en el que están los ficheros que ya existían en el disco original más un CONFIG.SYS cuyo contenido es «COUNTRY = 034» y un AUTOEXEC.BAT cuyo contenido es «KEYBSP». A partir de ese momento, al arrancar con el disco así creado el formato de la fecha y las características del teclado se adaptarán al idioma castellano.

El comando SHARE permite instalar una serie de programas residentes que se encargan de gestionar aquellas instalaciones en las que se compartan ficheros, como puede ser el caso de tener varios ordenadores conectados en red local.

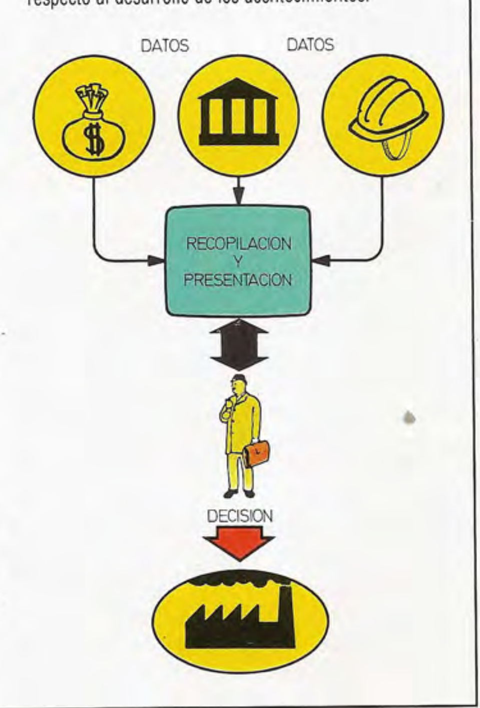
El comando DEVICE es uno de los que deben aparecer en el fichero de configuración CONFIG.SYS. Su misión es instalar drivers en la parte superior de la zona de memoria disponible con funciones tales como gestión de secuencias de escape, instalación de discos virtuales, etc. Esta última opción es la más interesante de todas. Un disco virtual consiste en reservar una zona de memoria central de forma que actúe como si de un disco ordinario se tratara. Cuando



Los progresos en microelectrónica y en la construcción de unidades de disco flexible de bajos perfiles y consumo, han puesto al alcance de la mano equipos con potencias idénticas a las de los ordenadores de sobremesa, pero que tienen la facultad de poder ser trasladados sin grandes inconvenientes hasta el punto justo en el que son necesarios.

El ordenador como ayuda en la toma de decisiones

Una de las aplicaciones más espectaculares del ordenador es la de ayudar al usuario en la toma de decisiones. Partiendo de un conjunto de datos, y aplicando criterios previamente definidos y plasmados en un programa, el ordenador elabora y presenta los datos bajo distintas perspectivas y muestra las alternativas a que conducen la modificación de ciertos datos. Son múltiples los ejemplos cotidianos que pueden ilustrar este ámbito de aplicación del ordenador. Por ejemplo, en el entorno empresarial, es posible, a partir de los diferentes costes pormenorizados de una cadena de producción, analizar las variaciones en el precio final de los productos; variaciones resultantes de cualquier fluctuación en los elementos que determinan el coste en la etapa de producción. Y todo ello en un breve espacio de tiempo.
Otro ejemplo, algo más extremo, cabe encontrarlo en los métodos para la defensa del espacio aéreo de un país. En este caso, partiendo de los datos suministrados por elementos como el radar y observadores humanos, el ordenador analiza la situación que se crea. Para ello, asigna prioridades a los diversos aparatos enemigos en función de su distancia a puntos neurálgicos del país y de la velocidad con la que se acercan a los mismos. Finalmente, presenta al responsable un cuadro de la situación, de manera que este último pueda decidir la zona en la que debe concentrar sus fuerzas. Todo el proceso se realiza en un tiempo mínimo, permitiendo que las decisiones se tomen de manera casi instantánea respecto al desarrollo de los acontecimientos.



bajo MS-DOS 3.00 se arranca con un disco en el que existe un fichero CONFIG.SYS entre cuyo contenido se encuentra la línea «DEVICE = VDISK.SYS 360», se crea automáticamente un disco virtual con capacidad para 360 Kbytes que será referido como C: en un sistema de dos unidades. El trabajo con esta nueva unidad es en todo punto idéntico al modo de actuar con el resto de las unidades convencionales. Su elevada velocidad de operación (superior a la de un disco rígido) tiene como contrapartidas la disminución del área de memoria donde se ejecutan normalmente los programas de aplicación, pudiéndose llegar a dar el caso de que ciertos programas no «quepan» en la nueva cantidad de memoria disponible, y el hecho de que su contenido se pierde completamente en el momento de que se desconecte la alimentación del equipo.

El comando FCBS permite especificar el número de «File Control Blocks» —Bloques de control de ficheros— que pueden estar en un momento determinado abiertos por el DOS. Este comando ha de aparecer en el interior del fichero CONFIG.SYS. El número de FCBs está en relación directa con el máximo número de ficheros que el sistema puede tener abiertos en un determinado instante. Normalmente, en los manuales de los programas de aplicación se encuentra la información necesaria para la modificación de este número, si bien la mayoría de las veces esta modificación no es necesaria y basta con no incluir el comando en el fichero de configuración para que el sistema adopte unos determinados valores por defecto suficientes para programas de aplicación sin grandes exigencias.

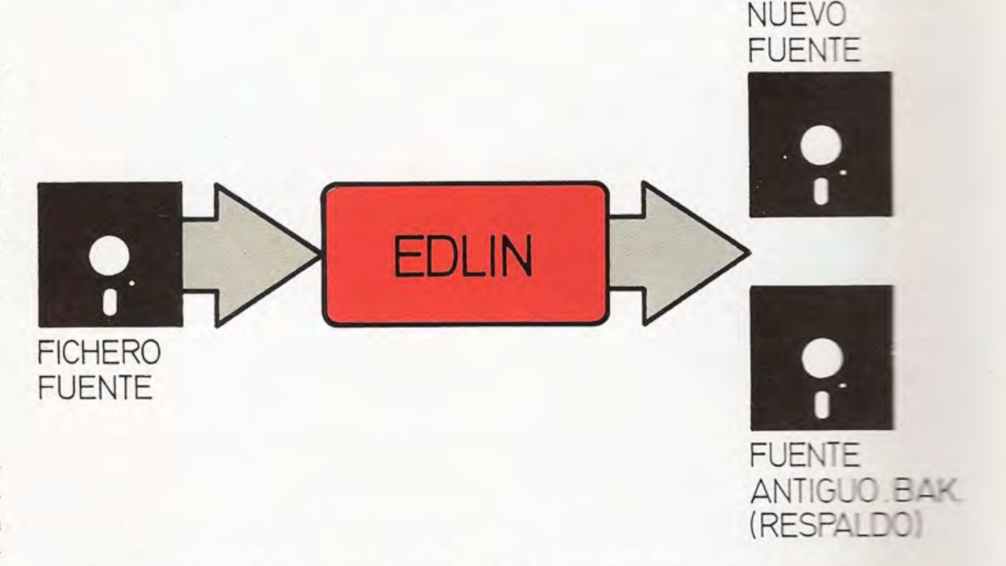
Para finalizar, con la versión 3.00 aparece el comando LASTDRIVE, cuya misión es ajustar el máximo número de unidades de disco a las que se puede acceder. Su sintaxis es «LASTDRIVE = x», en donde «x» puede ser cualquier carácter desde la A a la Z, y su aparición sólo puede ocurrir en el interior de un fichero CONFIG.SYS. En el caso de que no aparezca, por defecto la «x» toma el valor «E». El valor mínimo que puede tomar el parámetro de LASTDRIVE es el número de unidades físicas que están conectadas al sistema. En el caso de que «x» sea menor que la citada cantidad, el comando es ignorado.

MS-DOS (3)

El punto de vista del programador



Cuando el usuario se dispone a preparar sus programas, generar ficheros de datos o de texto, o cualquier otra tarea relacionada con el mantenimiento intrínseco de la información contenida en los ficheros, se ve obligado a recurrir al denominado editor.



Tras realizar su función, el editor del MS-DOS conserva siempre una copia de seguridad o de respaldo (backup) de la información previa.

El editor

Tal es el nombre que recibe la herramienta básica para este tipo de operaciones. En el mercado actual existen un gran número de ellos, cada uno con distintas características y posibilidades. Algunos, como los procesadores de textos, son editores de gran potencia, si bien resultan excesivos para tareas tales como el mantenimiento de programas o de simples ficheros de texto; de ahí que para el cometido señalado se emplee el editor que viene asociado al sistema operativo.

El editor del MS/DOS se presenta a modo de comando externo, invocable por medio de la orden EDLIN. Concretamente se trata de un editor de líneas, de ahí su denominación (EDitor LINe), por contraste con otros editores denominados de pantalla completa o «full screen», los cuales operan sobre el contenido total de la información visualizada en cada instante en la pantalla. Ello no implica, sin embargo, que este editor sea de peor calidad sino, sencillamente, que está orientado a trabajar línea a línea.

Una característica importante del editor del MS/DOS es que comparte su método de operación, por lo que al almacenamiento de información se refiere, con el editor del sistema operativo CP/M. Así, por ejemplo, no elimina la versión antigua de un fichero, sino que la almacena con el mismo nombre, aunque caracterizándola con la extensión «BAK». A su vez, la nueva versión toma el nombre original del fichero fuente. Ello obliga al usuario a tomar las adecuadas precauciones con los nombres de los diferentes ficheros, con objeto de que la distinción entre ellos no se reduzca exclusivamente a la extensión del nombre.

Semejante procedimiento tiene el inconveniente de que se duplica la información almacenada y si se editan muchos ficheros puede llegar a ocuparse con rapidez el soporte de almacenamiento.

- *Operación con el editor*

Cuando se invoca al editor, hay que especificar sobre qué fichero se va a actuar, siguiendo el procedimiento que se indica a continuación:

EDLIN A:DATOS.TEXT

Tras esta operación, el editor inicia un proceso de búsqueda y carga en memoria del fichero solicitado, brindando al usuario distintas respuestas de acuerdo a la situación que se plantee:

a) Si el fichero no resulta localizable,



La presencia de un sistema operativo dotado de potentes herramientas de edición y puesta a punto de programas, contribuye a distanciar a los equipos profesionales de los ordenadores domésticos.



El editor del MS-DOS cae dentro del grupo de los llamados «editores de líneas»: su trabajo se realiza sobre una sola línea en cada momento. Entre los comandos de edición se encuentra el de borrado, invocado a través de la tecla D.

bien porque sea nuevo y deseemos crearlo, o bien porque se haya introducido un nombre erróneo, aparecerá en la pantalla el mensaje de «nuevo fichero»:

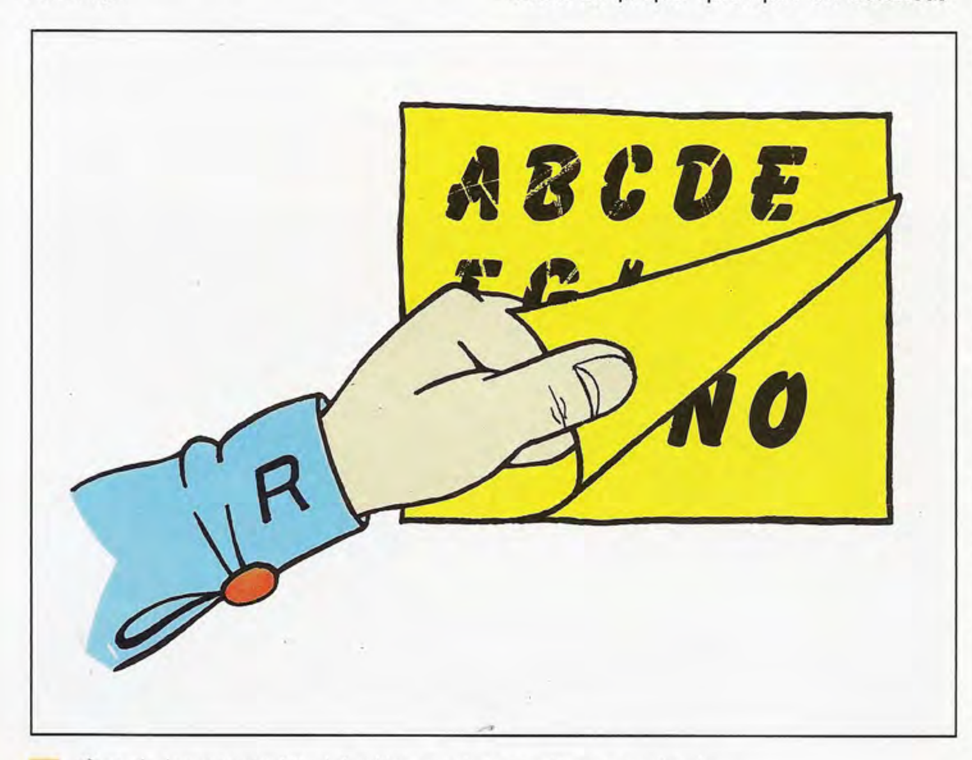
New File

b) Si se encuentra el fichero en cuestión, y su tamaño es adecuado para la carga de memoria y posterior operación con el mismo, la pantalla ofrecerá el mensaje:

End of Input File

c) Puede ocurrir que el fichero llegue a localizarse pero que su tamaño sea superior al de la memoria disponible. En tal caso no aparecerá mensaje alguno y el editor no cargará en memoria todo el fichero, sino tan sólo una parte del mismo, procediendo a actualizaciones sucesivas de su contenido.

Tras este proceso, el editor queda listo para operar, cosa que indica presentando su propio «prompt»: un asterisco (*). A partir de ahí, el usuario tiene a su disposición el conjunto de comandos del editor, entre los que se señalan a continuación los más importantes.



Otro de los comandos del editor es el invocado por medio de la tecla R. Su misión consiste en la búsqueda y sustitución de un determinado grupo de caracteres por otro.

El comando E se encarga de dar por terminada la tarea de edición, y activa el proceso de creación del nuevo fichero fuente y del fichero de respaldo. El comando Q también finaliza la edición, si bien el resultado no afecta a los ficheros fuente y de resplado; por lo cual, el trabajo realizado no se refleja en el fichero cuyo contenido se ha editado.

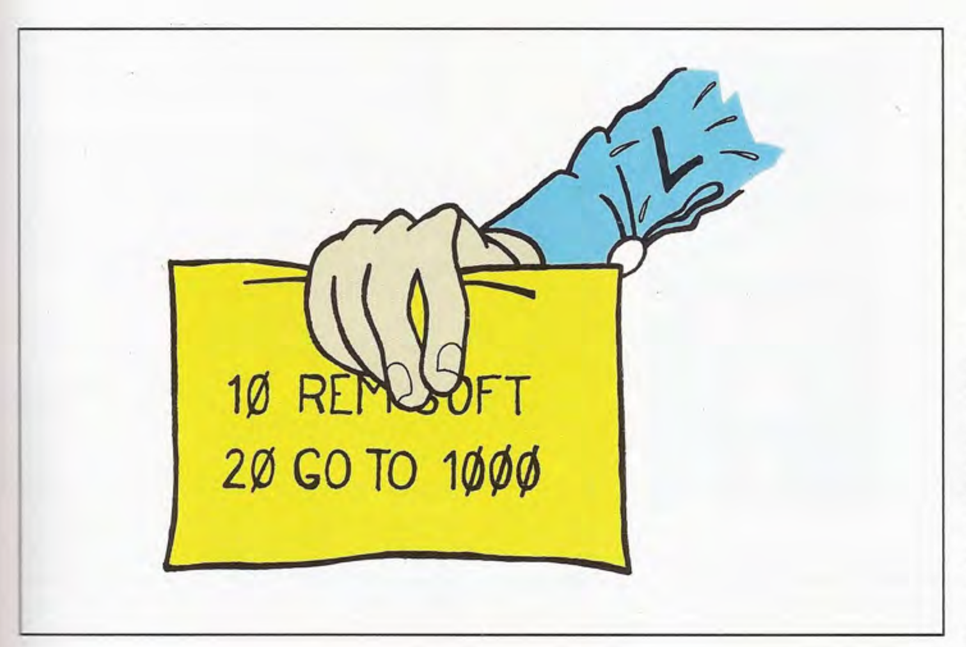
Las funciones de desplazamiento a través del fichero y de inserción y borrado de la información, se realiza por medio de tres comandos básicos: I (Insertar), D (Borrar) y L (Listar). Una vez seleccionada la línea a partir de la que se desea introducir información, la activación del comando I permitirá la inclusión de sucesivas líneas de datos o de programa. Cada vez que se introduce una nueva línea y se pulsa la tecla RETURN o ENTER, el sistema la almacena en el buffer del editor y queda en disposición de aceptar otra línea.

Cuando se desea crear un fichero nuevo, éste será el primer comando que debe entrar en acción. Al finalizar la entrada de nueva información y con el fin de regresar al estado normal de operación del editor, es necesario pulsar la tecla BREAK o <crtl -C>. Hecho esto, aparece de nuevo el prompt del editor (el asterisco). El comando D resulta adecuado para borrar a voluntad una o varias líneas de información. La forma que puede tomar esta orden es variable, aunque las más importantes son las que se relatan a continuación. Si se pulsa D sin añadir cualquier otro parámetro, será borrada la línea con la que se opera en ese preciso instante. Por el contrario, si se especifica un número de línea, será ésta la afectada por el borrado. Para eliminar un bloque de líneas de una sola vez, hay que optar por uno de los dos procedimientos que siguen. El primero adopta la formulación:

<nº línea inicial>,<nº línea final> D

Ahora las líneas eliminadas son las delimitadas por los números señalados en el comando.

El comando L permite al usuario desplazarse de un punto a otro del fichero en edición para ver su contenido y mo-



Cuando el programador desea visualizar una línea o un conjunto de líneas del fichero en edición, debe recurrir al comando L en una de sus múltiples variantes.

dificarlo si fuera necesario. Su modo de operación es similar al propio del comando Delete (D).

Si se invoca simplemente pulsando la tecla L, aparecerán en la pantalla un total de 23 líneas de texto delimitadas por la línea en curso de edición; esto es: localizadas por encima o por debajo de ésta. No obstante, de especificar un número de línea en la forma:

<nºlínea>L

se visualizarán las 23 líneas, si bien, la central corresponderá con la línea especificada. Esta última formulación es la que permite desplazarse al punto del fichero que interese visualizar; desde luego, este punto no tiene por qué coincidir con la línea actual en operación. También es posible por medio de este comando visualizar un determinado bloque de líneas, precisando los números de las líneas inicial y final del mismo:

<nºlínea inicial>,<nºlínea final>L

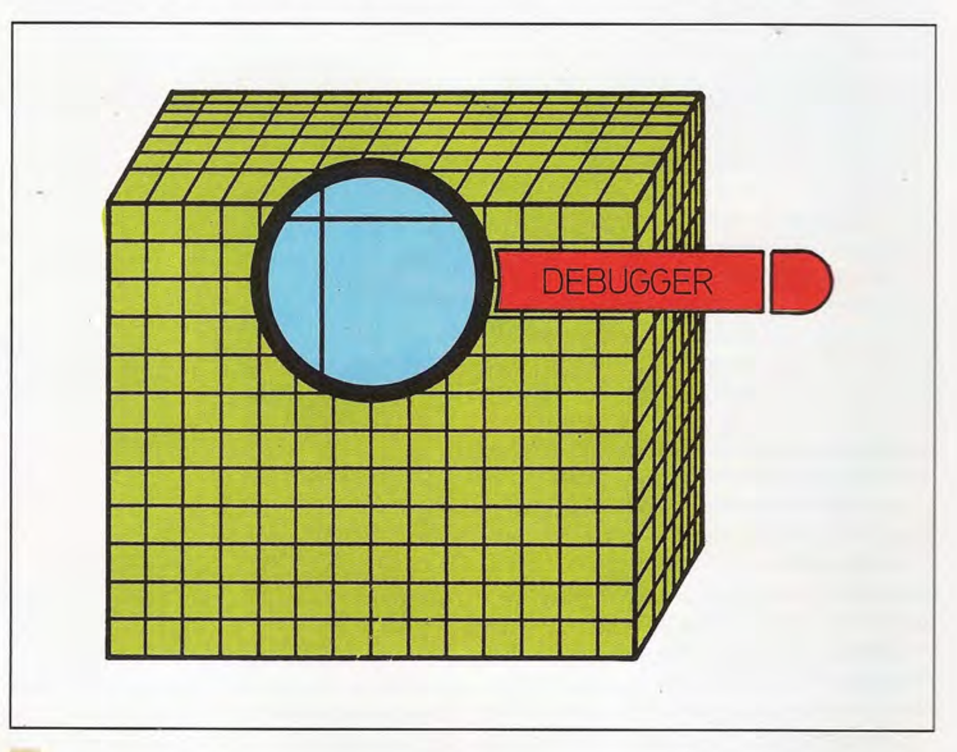
Por último, cabe mencionar a dos comandos de gran interés para el usuario, ya que permiten la localización rápida de determinados conjuntos de caracteres a los que el programador o el usuario desea acceder directamente, para examinarlos o modificarlos. Estos comandos responden a los apelativos S (búsqueda) y (R) sustitución.

El comando S se utiliza para ordenar la búsqueda de un determinado grupo de caracteres en el fichero; por ejemplo, una determinada palabra. El formato más general del referido comando es el siguiente::

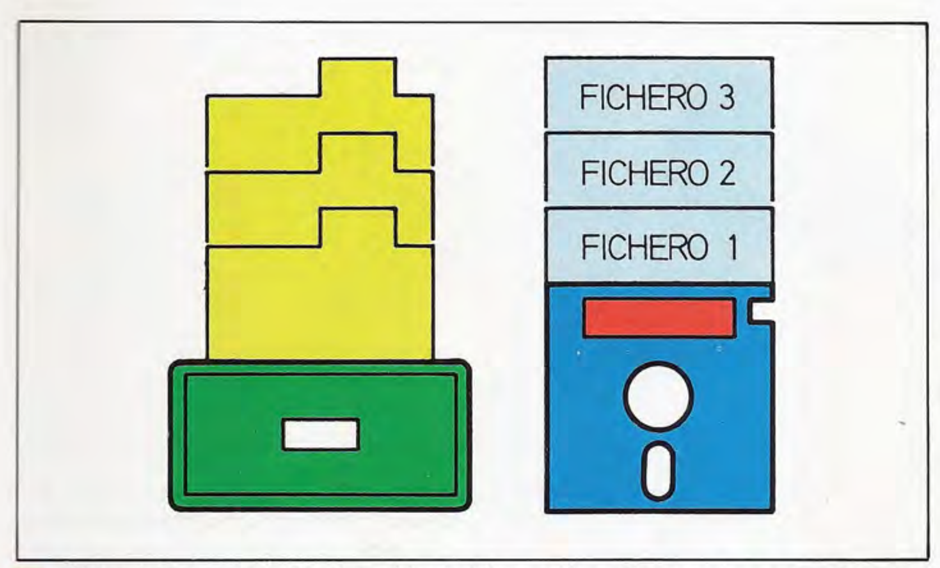
<nºlínea inicial><nºlínea final>S
<caracteres a localizar>

Bajo este formato se producira la búsqueda de los caracteres a localizar entre las líneas señaladas. Cada vez que se detecte su presencia, el ordenador mostrará la línea en la que se encuentran. Tras iniciar la búsqueda, bastará con pulsar S, sin introducir ningún tipo de parámetro adicional, para que ésta prosiga automáticamente saltando, de grupo de caracteres en grupo de caracteres.

El comando R opera de forma análoga, con la única diferencia de que hay que especificar otro grupo adicional de caracteres, el cual sustituirá al inicialmente señalado como objetivo de la búsqueda. Una vez iniciado el proceso, cada vez que se localice el referido grupo de caracteres se interrumpirá la busqueda y la máquina interrogará al usuario para conocer si éste desea que los



El «debugger» es una herramienta del sistema operativo MS-DOS orientada a la corrección y puesta a punto de programas.



Al igual que en un cajón de archivo la información está separada en carpetas, en un disco magnético se encuentra distribuida en ficheros.

caracteres sean sustituidos por los nuevos o no.

Debugger

El «debugger» es una poderosa herramienta que el MS/DOS pone a disposición del usuario especializado para la corrección y puesta a punto de sus programas. Como tal, está destinada al programador y no al usuario cuyo único deseo es la explotación de paquetes de aplicación ya ultimados. Con la ayuda del «debugger», es posible acceder a las distintas posiciones de memoria del ordenador, consultar su contenido y, de esta forma, comprobar si los resultados intermedios del programa son correctos; en el caso de que estos no sean correctos, el programador estará en condiciones de precisar a partir de qué punto empiezan a producirse los fallos, lo que facilitará en gran medida su corrección.

Los detalles operativos del «debugger» son sumamente dependientes del microprocesador que constituye el cerebro de cada máquina, de ahí que no entremos en detalles sobre el mismo. No obstante sí es oportuno precisar que el «debugger» del MS/DOS opera con aritmética hexadecimal y, por ello, el contenido de las posiciones de memoria examinadas puede aparecer ante el usuario bajo un aspecto críptico. Hay que tener en cuenta además, que el «debugger» revelará, por medio de abreviaturas, cuál es la naturaleza o tipo de la información almacenada en cada posición de memoria.

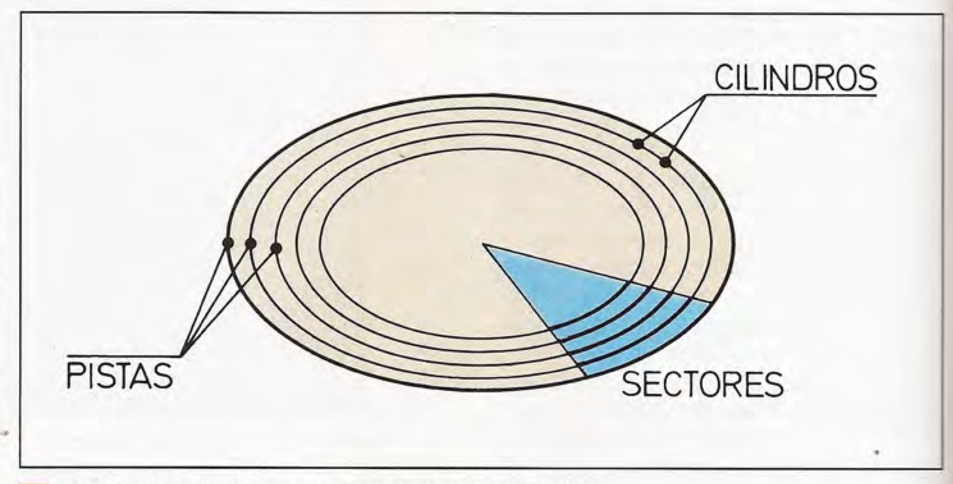
Gestión del almacenamiento en discos

El almacenamiento de datos para su conservación y uso posterior, supone el empleo de unidades de memoria externa. Entre éstas, las más ampliamente utilizadas en el mundo de la informática son las que utilizan discos magnéticos, flexibles o rígidos, como soporte de almacenamiento.

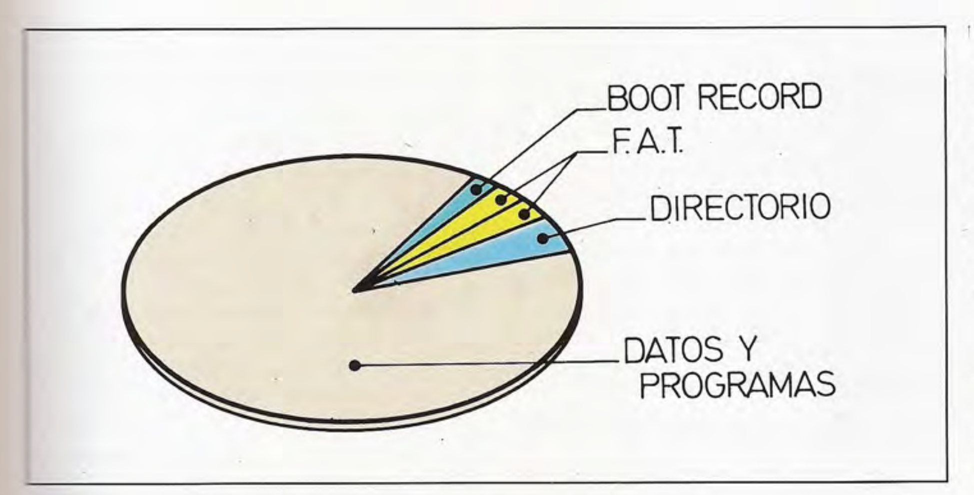
En el disco, la información está organizada en ficheros de diverso contenido. Algo semejante a lo que ocurre con las carpetas alojadas en un cajón de archivo, cada una de ellas destinada a acoger información de diversa índole.

Todo dispositivo de almacenamiento tiene una capacidad nominal, expresada en Kilobytes (Kb) para los disquetes o discos flexibles y en Megabytes (Mb) en el caso de los discos rígidos. Esta medida indica la cantidad teórica de información que pueden contener. Sin embargo, no todo este espacio es accesible al usuario. El dispositivo que controla el disco o disquete precisa cierta información para su manejo; información que debe ser también almacenada y para lo cual reserva parte del espacio disponible.

Esto es lo que sucede cuando se formatea un disco virgen en orden a permitir su uso posterior. Si no se realizara el proceso de formateo o inicialización, el disco no sería reconocible por el sistema, al no contener ninguna información identificativa. La información residente en un disco o disquete se agrupa en bloques o conjuntos de bytes susceptibles de ser leídos o escritos con una sola operación de lectura o escritura. A este conjunto de bytes se le denomina *sector* y su tamaño varía según sea la densidad de almacenamiento que es ca-



La distribución física del disco magnético se realiza en sectores; estos, a su vez, se agrupan en pistas, componiendo estas últimas los cilindros.



El espacio de almacenamiento del disco no está enteramente disponible para los ficheros que desee almacenar el usuario. Una determinada zona del mismo queda reservada para uso del sistema y para el dispositivo de control de los discos.

paz de soportar la superficie del disco. Son típicos, por ejemplo, los sectores de 512 bytes para discos flexibles de una cara y simple densidad. La siguiente etapa a la hora de construir la estructura de la información almacenada en un disco, es la agrupación de sectores hasta cubrir la longitud de un círculo completo. A esta agrupación de sectores se la conoce como *pista* o «track»; en la mayoría de los formatos existentes en el mercado, la pista está integrada por ocho sectores.

Si existe más de una superficie en el disco o disquete sobre la que es posible grabar la información (discos de doble cara), nace el concepto de *cilindro,* que se incluye a la pista existente en la cara superior y a la localizada en la cara inferior. Al respecto, cabe dar la definición de cilindro como «conjunto de pistas que se encuentran a la misma distancia del centro».

Como consecuencia de su definición, los sectores situados en el mismo cilindro pueden ser leídos o escritos sin necesidad de desplazar el mecanismo de lectura/escritura; tan sólo hay que esperar a que el giro del disco ponga el sector afectado al alcance de la cabeza de lectura/escritura. En el caso de que para realizar la lectura o escritura fuese necesario el desplazamiento de la cabeza, el tiempo que tardaría ésta en posicionarse en el lugar correcto sería relativamente elevado; en consecuencia, la búsqueda sería lenta en comparación con la lectura/escritura en el mismo cilindro.

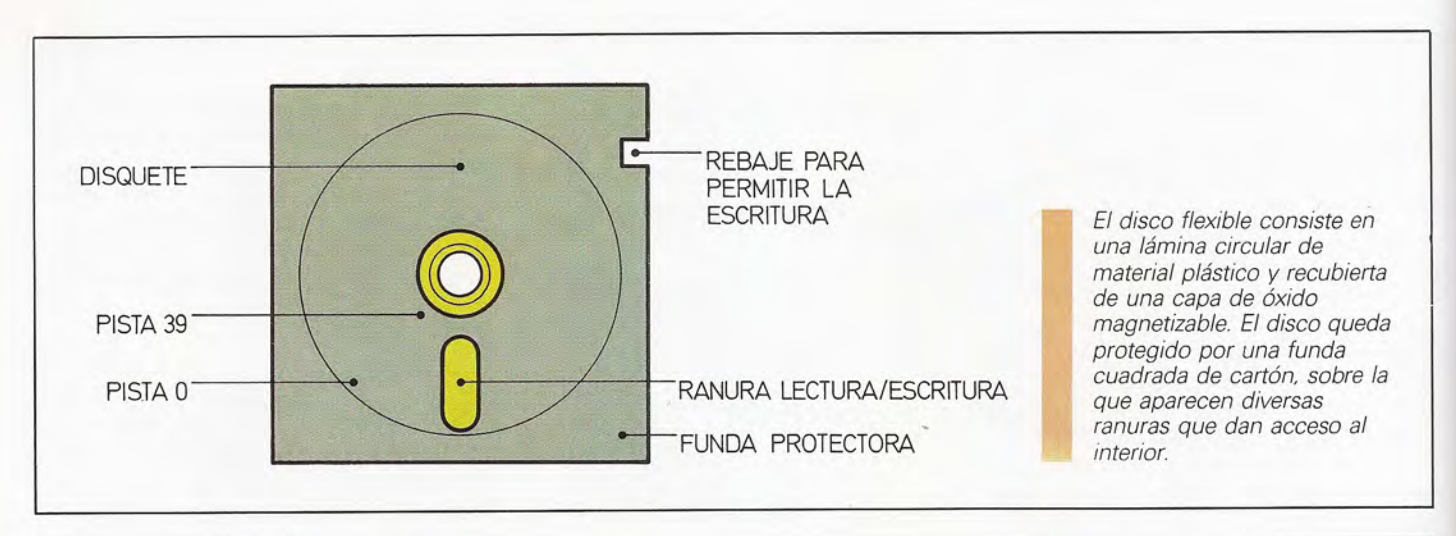
El número de pistas típico en un disquete suele ser de 40 a 80 por cada superficie activa del mismo. A su vez, los discos rígidos suelen estar organizados en unos 305 cilindros cada uno. Una vez que se han definido los diversos conjuntos de bytes en los que está organizado el disco, hay que examinar cuál es la disponibilidad del espacio así definido. Como se indicó al principio, no todo el espacio nominal puede ser utilizado por el usuario, sino que una porción del mismo queda reservado para el uso del sistema. En términos generales, el espacio se distribuye de la siguiente forma:

— Registro de arranque o «boot record».
— Primera copia del archivo que contiene la tabla de asignaciones de espacio.
— Segunda copia del archivo que contiene la tabla de asignaciones.
— Directorio.
— Area de datos.

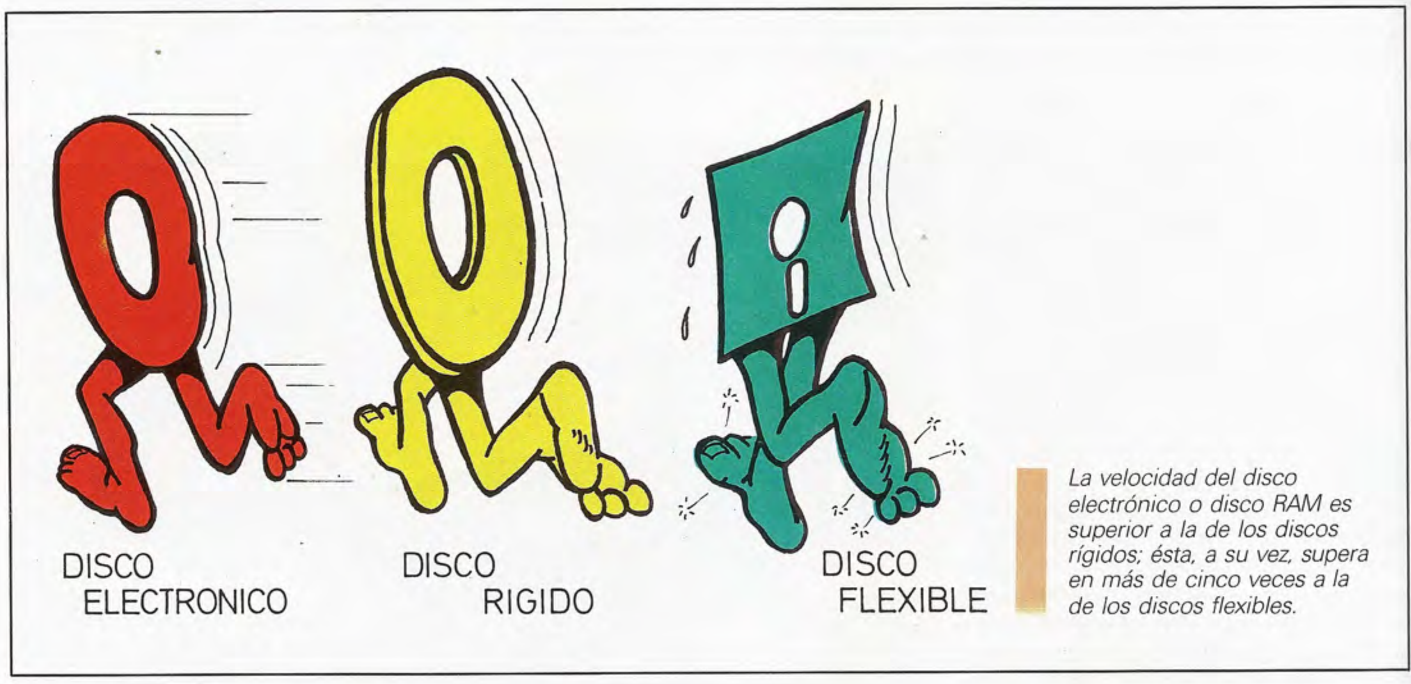


En los ordenadores que disponen de disco rígido, este suele orientarse a las operaciones de almacenamiento en régimen de trabajo. En tal caso, los discos flexibles, dada su menor capacidad y velocidad de acceso, se reservan para la obtención de copias de seguridad del contenido de los discos rígidos.

El registro de arranque contiene un programa, de reducida longitud, cuyo cometido es ayudar a la puesta en marcha del sistema operativo MS/DOS desde el disco. Este forma parte de todos los discos, aunque la mayor parte de ellos no albergan al sistema operativo sino tan sólo ficheros de datos.



El disco flexible consiste en una lámina circular de material plástico y recubierta de una capa de óxido magnetizable. El disco queda protegido por una funda cuadrada de cartón, sobre la que aparecen diversas ranuras que dan acceso al interior.



La velocidad del disco electrónico o disco RAM es superior a la de los discos rígidos; ésta, a su vez, supera en más de cinco veces a la de los discos flexibles.

Los espacios del disco utilizados están registrados en todo momento en el fichero que contiene la tabla de asignaciones de espacio o FAT (del inglés File Allocation Table). Dicha tabla memoriza dónde está localizado cada fichero y qué zonas del disco permanecen libres.

El directorio del disco es una tabla en la que reside una escueta información de cada uno de sus ficheros. Tal información incluye el nombre de cada fichero y el nombre de su extensión, el tipo del fichero, la fecha y hora en el que fue creado o modificado por última vez, su posición inicial dentro del disco y su longitud en bytes.

Discos flexibles

Este medio de almacenamiento es el más frecuentemente utilizado en el campo de los microordenadores debido, básicamente, a su coste asequible y a las cada vez más altas prestaciones. Si bien, empiezan ya a ser reemplazadas por nuevos dispositivos más avanzados, hasta ahora exclusivos de ordenadores mucho más potentes.

Los discos flexibles, disquetes, o «floppies» están fabricados en plástico flexible. Ello permite que éste sea un medio de almacenamiento poco exigente para su conservación, facilitando su transporte y almacenamiento.

El disquete está constituido por una funda protectora de forma cuadrada, de plástico o cartón rígido, en la que aparecen varias ranuras y agujeros. El agujero central tiene por función dejar al descubierto la zona del disquete más próxima al centro, para que el mecanismo de arrastre sea capaz de sujetar al disquete y hacerlo girar. Dicha zona no almacena información alguna, ya que podría ser dañada fácilmente.

La ranura existente debajo del agujero central permite que se produzcan a través de ella las diversas operaciones

de lectura/escritura, a realizar por la cabeza al efecto. En último término cabe mencionar la ranura existente en uno de los lados, en forma de pequeño entrante cuadrangular, que permite inhibir la escritura en el disquete si dicha zona aparece cubierta.

Los disquetes más comunes son los de 5 y 1/4 pulgadas, aunque actualmente empiezan a proliferar los de 3 y 1/2 pulgadas. Desde luego, también se utilizan los de ocho pulgadas, sobre todo en equipos antiguos.

Otra característica a destacar es la densidad de la información grabada en el disco: simple, doble y cuádruple densidad con 24, 48 y 96 pistas por pulgada, respectivamente. Por supuesto, el disco puede estar grabado por una sola cara o ambas.

Respecto a la posibilidad de formateado de los disquetes, cabe señalar que puede realizarse bajo el control de un programa al efecto, aunque también hay discos cuya distribución o formato queda en el proceso de fabricación. En el primer caso, la sectorización es de tipo «soft» (soft-sectored), mientras que en el segundo hay que hablar de discos «hard-sectored». Por último, la capacidad de almacenamiento tiene un amplio margen de variación: desde los 100 Kbytes hasta algunos Mbytes.

Discos rígidos

Los discos rígidos están fabricados en base a un soporte rígido recubierto de material magnético. Una de las tecnologías más frecuentes es la denominada Winchester; de ella procede el uso extendido de llamar discos Winchester a la generalidad de discos rígidos.

Su capacidad de almacenamiento suele estar comprendida entre los 5 y los 50 Megabytes, con lo que sobrepasan ampliamente la capacidad de los disquetes. Los discos rígidos están sellados, debido a que la separación entre la superficie del disco y las cabezas lectoras es tan pequeña que cualquier partícula de polvo podría quedar atrapada en el espacio intermedio, pudiendo dañar la superficie magnética del disco.

Así pues, los discos no pueden ser extraídos fuera del mueble que los aloja, contrariamente a lo que ocurre con los discos flexibles. Pierden su movilidad, aunque su gran capacidad de almacenamiento hace que no sea necesaria su sustitución por otro disco vacío en la mayor parte de los casos. No obstante, existen discos rígidos contenidos en un

Ficheros para el almacenamiento de datos

Una de las bases de la informática reside en la posibilidad de almacenar información, de tal manera que pueda ser posteriormente recuperada y utilizada en otros procesos. El almacenamiento afecta tanto a los programas cuya misión es procesar los datos, como a los propios datos manipulados.

Un *fichero* de datos está constituido por un conjunto de *registros,* los cuales acogen los datos distribuyéndolos internamente en una serie de subdivisiones denominadas *campos.* Como ejemplo, cabe imaginar a un registro como la porción del archivo que almacena el conjunto de datos correspondientes a una persona: su nombre, apellidos y dirección; mientras que los campos serían las porciones del registro que contienen los valores asociados a cada uno de los tres conceptos.

Los ficheros pueden clasificarse en muy diversas categorías, entres las que cabe destacar a los ficheros de tipo secuencial, de acceso directo y multi-indexados.

En síntesis, un fichero es de tipo secuencial cuando para acceder a la información contenida en uno cualquiera de sus registros es necesario ir leyendo, uno tras otro y desde el principio, hasta llegar al registro que contiene la información buscada.

Obviamente, este método no contribuye en absoluto a favorecer el rápido acceso a la información.En un fichero de acceso directo, cada registro tiene asociado un número de identificación, y el acceso a la información en el contenido se efectúa a través del número que identifica al registro. Este método de acceso resulta muy conveniente cuando no se accede a más del 20 % de los registros del fichero. Los ficheros multi-indexados funcionan de forma muy similar a los de acceso directo, con la salvedad de que, en este caso, el elemento identificador no es un número exclusivo y propio de cada registro, sino que coincide con la información contenida en uno a varios de sus campos. Estos campos que se utilizan como identificadores son los que se denominan claves.

REGISTRO N
REGISTRO 2
REGISTRO 1

SECUENCIAL



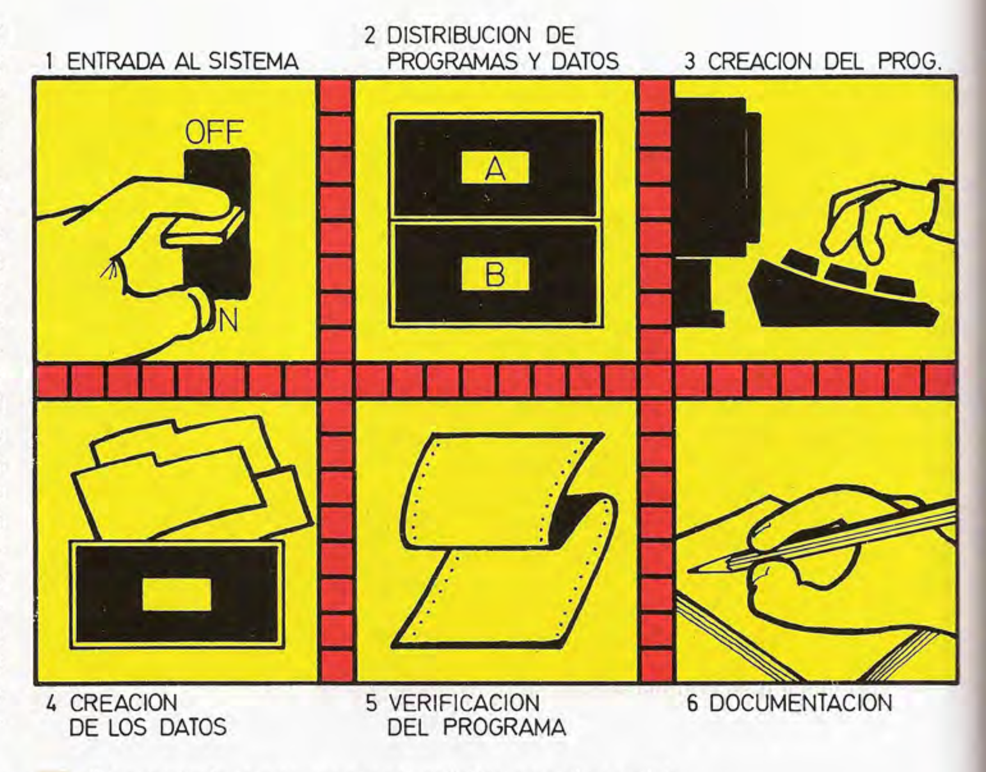
ACCESO DIRECTO



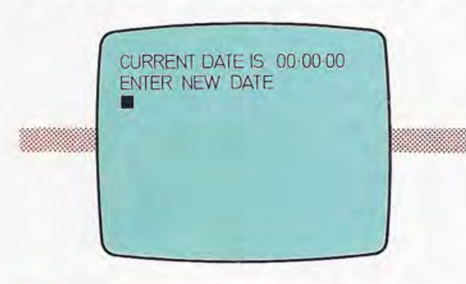
MULTIINDEXADO

cartucho que permite su extracción del dispositivo de control y la inserción en el mismo de un nuevo disco. Debido a su especial y delicado dispositivo de lectura/escritura, la velocidad de las operaciones sobre un disco rígido supera en más de cinco veces a la habitual en un disquete. Por lo que respecta a la distribución del espacio, hay que señalar que a la hora de definir el disco mediante el comando FDISK, se pueden asignar hasta un máximo de cuatro zonas diferentes llamadas particiones, pudiendo ser cada una de ellas de distinto tamaño. El formateo de cada una de las particiones también puede diferir. De esta forma es posible conseguir que una partición contenga a un sistema operativo y sus ficheros correspondientes, y en otra aloje a un sistema operativo diferente. Esto es: se puede formatear una zona apropiada para el MS/DOS, e incluir en ella este sistema operativo, mientras que otra se puede inicializar con formato CP/M-86 y almacenar el sistema operativo CP/M-86.

Las particiones vienen a ser compartimentos estancos, de tal forma que no es posible transferir directamente datos de una partición a otra y, desde luego, tampoco puede estar un fichero a caballo entre dos particiones.



Secuencia de pasos a poner en práctica para el completo y correcto desarrollo de un programa.



La introducción de la fecha y hora actuales en el ordenador es una medida muy recomendable. Ello permitirá al usuario contar con una información cronológica exacta relativa a la creación o última modificación de los archivos manipulados.

Microdiscos y discos electrónicos

Los microdiscos, en sus formatos de 3" y 3 y 1/2 pulgadas —básicamente estos últimos—, están entrando con fuerza en el terreno de la microinformática. Su presencia es cada vez más común en equipos con MS/DOS. Los microdiscos aparecen encerrados en una caja de plástico rígido, en la que existe una trampilla que pone al descubierto una porción de superficie magnética. Otro tipo de disco, de reciente actualidad, es el llamado disco electrónico o disco RAM. Estrictamente, no se trata de un sistema para el almacenamiento de información, sino de una combinación de memoria y programas que permiten simular el funcionamiento de un disco en memoria RAM.

El motivo principal que aboga por la utilización de este tipo de dispositivos, reside en la gran diferencia de velocidad existente entre la CPU y las unidades de disco; ello hace que se produzca un cuello de botella en la velocidad de ejecución cuando un programa se ve obligado a acceder reiteradamente al disco. Con un disco electrónico se pueden mover datos desde un disco convencional hasta el disco electrónico, de manera que las operaciones de lectura y escritura de los datos durante el proceso de ejecución se realicen a mayor velocidad.

Debido a su naturaleza no se trata exactamente de un sistema para almacenar información, sino para tratarla. Al desconectar el suministro de corriente, los datos almacenados se pierden, siendo por tanto un sistema de memoria volátil. Por ello, el método de trabajo con el disco electrónico se concreta en los siguientes puntos:

— Los datos se transfieren desde un sistema de almacenamiento permanente (disco o disquete) al disco electrónico.

— Se realizan las operaciones oportunas con los datos en el disquete electrónico.

— Una vez terminado el proceso, los datos vuelven a ser memorizados en un disco o disquete.

Desarrollo de programas

El desarrollo de programas por parte del usuario es una faceta a la que quizá no se preste la atención que merece, debido a la existencia en el mercado de

una gran cantidad de programas que cubren casi todas las necesidades (hojas electrónicas, tratamientos de texto, paquetes gráficos, gestión de archivos y bases de datos, contabilidad, etc.). De ahí que la tarea de desarrollar programas se reduzca, por lo general, a unas cuantas aplicaciones sencillas y muy específicas, para el uso particular de quien acomete su diseño.

En cualquier caso, estas aplicaciones suelen elaborarse empleando métodos poco disciplinados y ortodoxos; en parte debido a que suelen estar pensados para uso muy personal.

Aunque el volumen de programas de creación propia no acostumbre a alcanzar un nivel considerable, siempre es buena técnica disponer de una metodología con la cual llevarlos a cabo. Ello frenará el impulso de querer ultimar los programas inmediatamente después de sentarse tras el teclado.

A continuación se enumeran una serie de pasos a contemplar para el completo y correcto desarrollo de un programa: desde la entrada al ordenador, hasta la conclusión eficaz del programa y su posterior documentación. La descripción versará en torno a un ejemplo concebido para el ordenador personal IBM-PC XT, dotado del sistema operativo MS/DOS (PC/DOS) versión 2.00



La adopción de una metodología clara y sencilla a la hora de proceder al desarrollo de programas, reduce las probabilidades de error, facilita el trabajo, y contribuye a aprovechar con mayor eficacia los recursos que brinda el ordenador.

- *La entrada al sistema*

Como ya es sabido, al conectar el ordenador entra inmediatamente en funcionamiento un programa cuya misión es copiar el sistema operativo, desde el disco en el que reside, hasta la memoria del ordenador. Al finalizar este proceso se puede trabajar ya bajo el con-

Programas y datos

El sistema operativo sólo es capaz de distinguir dos tipos básicos de ficheros entre la multiplicidad de ellos que pueden residir en un disco: los ficheros de *programas* y los de *datos*. En un ajuste más fino, cabe distinguir ciertas particularidades dentro de cada uno de ambos grupos. En cuanto a los ficheros cuyo contenido son programas ejecutables, el MS/DOS distingue entre dos categorías: aquellos cuya extensión es COM y los de extensión EXE.

Los ficheros de tipo COM se utilizan para almacenar programas que están listos para su ejecución inmediata; su formato refleja la imagen que éstos tienen en la memoria del ordenador cuando el programa está en curso de ejecución. A su vez, los ficheros de tipo EXE son programas que requieren un último retoque antes de ser ejecutados; retoque necesario para que el programa sea cargado en la zona de memoria adecuada. Este último formato se utiliza con programas complejos y se obtiene como resultado de la compilación de dichos programas.

Desde el punto de vista del ordenador sólo existen dos tipos de programas. Un programa BASIC se considera como un tipo especial de fichero de datos, ya que su contenido no es ejecutable directamente. En efecto, hay que utilizar para ello un programa intérprete que a la vista de los datos contenidos en el fichero BASIC realice la acción oportuna.

Dentro de los ficheros de datos, merecen una mención especial los que están grabados con formato ASCII. En ellos, los caracteres se almacenan de acuerdo al código alfanumérico ASCII. La gran ventaja que aporta este tipo de ficheros de datos reside en su notable universalidad. El uso del código ASCII permite el fácil transporte de los datos contenidos en un fichero de un sistema a otro. De ahí que su empleo sea recomendable siempre que se prevea que los datos residentes en un fichero puedan ser de interés en otros sistemas.

DATOS — PROGRAMA — DATOS

trol del sistema operativo. Cabe recordar que es altamente recomendable disponer siempre de una copia de respaldo del sistema operativo para evitar disgustos en el caso de que el disco que lo contiene sufra daños irreparables. Acto seguido, el MS/DOS pedirá la fecha y hora con los formatos siguientes:

Current date is (DD-MM-YY):
00-00-0000
Enter new date
22-03-1985
Current time 0:0:0
Enter new time
10:00:00

(Los datos que figuran en cursiva corresponden a las respuestas que proporciona el usuario).

De no especificar la fecha y la hora, el sistema adoptará los datos horarios correspondientes al instante en que terminó la sesión anterior. En tal caso se perderán las ventajas de disponer como información adicional de la fecha y hora exactas de la creación o última modificación de los ficheros.

Antes de proseguir con el estudio, conviene recordar que los textos que aparezcan en letra cursiva, dentro de los ejemplos, serán introducidos por el usuario, e irán seguidos por una acción sobre la tecla «ENTER» o «RETURN».

Mejoras y novedades del MS/DOS versión 2.00

La aparición de las versiones 2.00 y sucesivas del sistema operativo MS/DOS han hecho que éste adquiera progresivamente una mayor potencia y versatilidad.

Un hecho frecuente es que la versión del sistema operativo que equipa al ordenador en el momento de adquirirlo, quede superada al cabo de un cierto tiempo por nuevas revisiones. En muchos casos, la adopción de una nueva versión del sistema operativo se retrasa excesivamente por lo tedioso que puede resultar la tarea de actualizar la configuración.

En este punto cabe señalar que el MS/DOS permite un cambio de configuración sencillo cada vez que se arranca el sistema. Ello se debe a que es un fichero editable el que incluye a los parámetros de configuración, tales como los nombres de los dispositivos, las características de los discos y otra información adicional relativa al modo de operación del MS/DOS.

Una diferencia sustancial que caracteriza al MS/DOS versión 2.00, queda reflejada en la posibilidad de trabajar con discos rígidos. Estos permiten un acceso mucho más rápido a la información almacenada, al tiempo que incrementan espectacularmente la capacidad de almacenamiento externo. En este mismo entorno, hay que citar la posibilidad de trabajar con directorios jerárquicos, organizando los ficheros en base a estructuras arborescentes. Con ello se logrará organizar la información de una forma más eficiente.

En cuanto a la flexibilidad de operación, cabe destacar el redireccionamiento de las salidas y entradas por pantalla y teclado a otros dispositivos asociados al sistema; la posibilidad de que un programa adopte automáticamente los datos de salida de otro programa como datos de entrada propios; el más eficaz control de la pantalla por medio de secuencias de caracteres especiales que controlan la posición del cursor; el trabajo con color; las funciones de las teclas definibles; etc.

Por último, hay que hacer mención a los nuevos comandos que aporta la versión 2.00 y a los ya existentes que se ven notablemente modificados en la nueva versión; todos ellos quedan reflejados en el cuadro adjunto.

COMANDOS MS/DOS	
Nuevos comandos del MS/DOS 2.00	Comandos modificados respecto a anteriores versiones
ASSIGN	CHKDSK
BACKUP	COMP
BREAK	DEBUG
CLS	DIR
CTTY	DISKCOPY
ECHO	DISKCOMP
IF	EDLIN
FOR	ERASE
SHIFT	FORMAT
GOTO	LPT2
FDISK	LPT3
GRAPHICS	COM2
MKDIR	
RMDIR	
CHDIR	
PATH	
PRINT	
PROMPT	
RECOVER	
RESTORE	
SET	
TREE	
VER	
VERIFY	
VOL	

- *Distribución de programas y datos*

En el caso de que se disponga de una revisión del MS/DOS coincidente o posterior a la 2.00, se puede aprovechar el uso de directorios y subdirectorios para estructurar mejor el espacio en disco, y almacenar agrupadamente los diversos tipos de ficheros. En este orden, es interesante poder definir varios directorios en los que se agruparán sólo programas, sólo datos, o sólo utilidades, etc...

En nuestro caso se creará un directorio para los programas (PROG) y otro para albergar los archivos de datos y resultados (DATA).

Así pues, mediante el comando MKDIR o MD se crearán los dos directorios antes citados. Previamente, seleccionaremos el disco introducido en la unidad A como disco de trabajo; el disco alojado en la unidad C quedará reservado a los programas de uso general. Las acciones a realizar son las ordenadas por el siguiente grupo de comandos:

C>*A:* (cambia la unidad de disco tomada por defecto)
A>*MD PROG* (crea el directorio PROG)
A>*MD DATA* (crea el directorio DATA)

El contenido del disquete puede examinarse apelando al comando DIR:

A>*DIR A:*
PROG <DIR>23-03-85 10:01:27
DATA<DIR> 23-03-85 10:01:30

- *Creación del programa*

Una vez creados los directorios, habrá

que dirigirse al directorio PROGR para crear en su ámbito el programa BASIC. El comando que permite cambiar de directorio es CHDIR o CD:

A>CD PROG

El contenido del directorio PROG es:

```
A> DIR
    .<DIR> 23-03-85 10:01:27
    ..<DIR>23-03-85 10:01:27
```

Los signos «.» y «..» son dos entradas que se utilizan para referenciar al directorio acutal (.) y al directorio raíz (..).

El programa BASIC se confeccionará con la colaboración del editor asociado al traductor BASIC; éste ofrece mayores ventajas al efecto que el editor de líneas EDLIN que acompaña al MS/DOS. Para invocar al BASIC bastará con ejecutar el siguiente comando:

A>C:BASICA

El programa cuyas sucesivas líneas habrá que introducir, puede coincidir, por ejemplo, con el que aparece en el cuadro adjunto. Su cometido se reduce a leer números de un fichero, ordenarlos y, posteriormente, escribirlos en otro fichero; ambos ficheros afectados se encuentran en el directorio DATA. Al terminar la introducción del programa a través del teclado, conviene almacenarlo en disco utilizando la orden:

OK SAVE "A: ORDENA.BAS",A

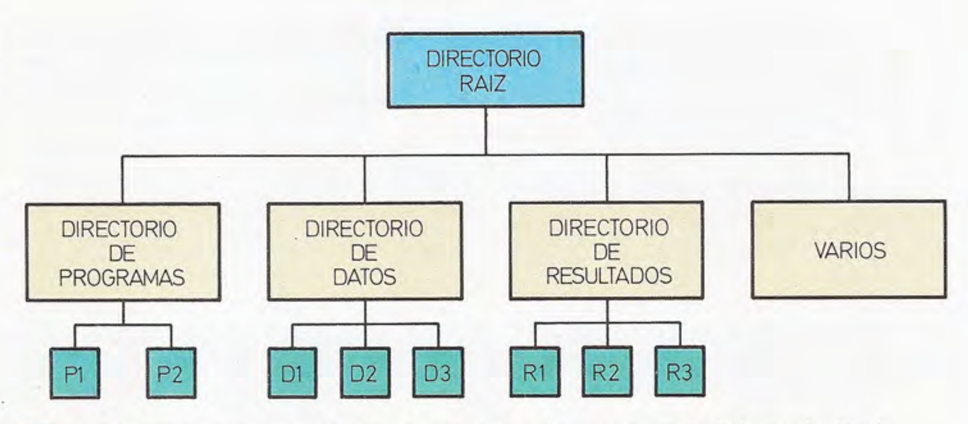
El parámetro «,A» especifica que la grabación se realice en el formato ASCII. Concluida la grabación, se regresará al control del sistema operativo apelando al comando al efecto:

OK SYSTEM

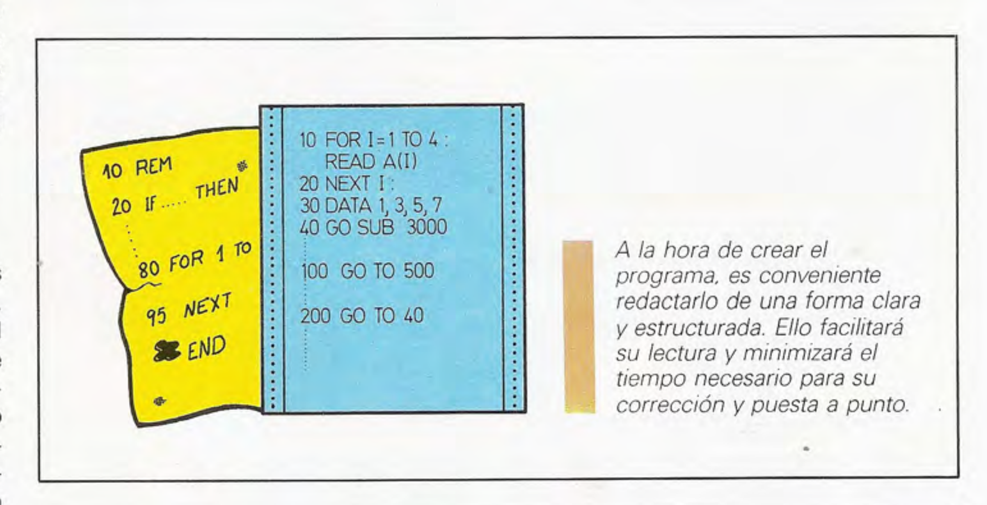
- *Creación de los datos*

Como se ha convenido anteriormente, los datos de entrada al programa estarán localizados en un fichero incluido en el directorio DATA. En consecuencia, es necesario dirigirse al directorio en cuestión:

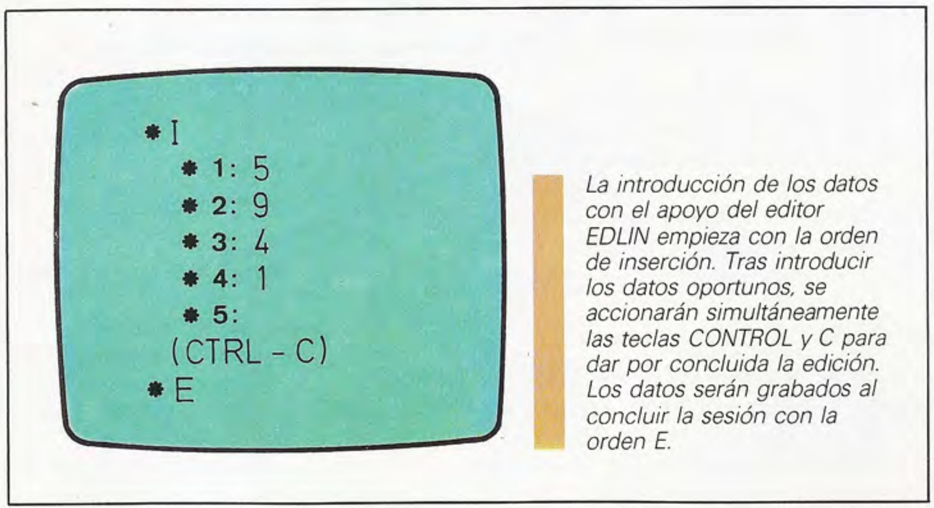
A> CD \ DATA



La compartimentación en directorios de los diversos tipos de ficheros facilita la tarea del mantenimiento y localización de las aplicaciones.



A la hora de crear el programa, es conveniente redactarlo de una forma clara y estructurada. Ello facilitará su lectura y minimizará el tiempo necesario para su corrección y puesta a punto.



La introducción de los datos con el apoyo del editor EDLIN empieza con la orden de inserción. Tras introducir los datos oportunos, se accionarán simultáneamente las teclas CONTROL y C para dar por concluida la edición. Los datos serán grabados al concluir la sesión con la orden E.

Por supuesto, también es posible editar directamente desde el emplazamiento actual, aunque tomando la precaución de indicar el camino que conduce al directorio DATA:

A> C:EDLIN \ DATA \ LIO.DAT

Por razones de simple comodidad, optaremos en nuestro ejemplo por este último camino; así se evitará la necesidad

```
10  REM PROGRAMA PARA ORDENAR NUMEROS
20  REM APERTURA DEL FICHERO DE DATOS Y DE RESULTADOS
30  OPEN "\DATA\LIO.DAT" FOR INPUT AS #1
40  OPEN "\DATA\ORDEN.DAT" FOR OUTPUT AS #2
50  REM LECTURA DE DATOS
60  INPUT #1, NPUN
70  DIM NUM(NPUN)
80  FOR I=1 TO NPUN
90      INPUT #1, NUM(I)
100 NEXT I
110 REM ORDENACION DE LOS DATOS
120 FOR I=1 TO NPUN-1
130     FOR J=I+1 TO NPUN
140         IF NUM(J) > NUM(I) THEN GOTO 180
150         PASO = NUM(J)
160         NUM(J) = NUM(I)
170         NUM(I) = PASO
180     NEXT J
190 NEXT I
200 REM ESCRITURA EN EL ARCHIVO DE RESULTADOS
210 WRITE #2, NPUN
220 FOR I=1 TO NPUN
230     WRITE #2, NUM(I)
240 NEXT I
250 REM CIERRE DE LOS FICHEROS
260 CLOSE #1
270 CLOSE #2
280 STOP
300 END
```

de saltar continuamente de un directorio a otro. Cabe observar que ahora se emplea el editor de líneas EDLIN, ya que en este caso no nos sirve el editor del BASIC.

Acto seguido, se introducirán los datos necesarios: cantidad de datos a ordenar y números que deben someterse al proceso de ordenación, uno por la línea de fichero. Una vez introducidos todos ellos se grabará el fichero en cuestión, tal y como ilustra la figura adjunta.

- *Verificación del programa*

En este momento se han creado ya los elementos necesarios para que sea posible llevar a cabo la ordenación. El siguiente paso debe coincidir, pues, con la ejecución del programa.

Habrá que invocar de nuevo al BASIC, para realizar la carga del programa y ejecutarlo a continuación; si bien, antes de ejecutarlo, se obtendrá un listado del programa para examinar su aspecto actual:

A> *C:BASICA*
OK *LOAD "ORDENA.BAS"*
OK *LIST*
    10 REM PROGRAMA PARA
    ORDENAR NUMEROS
    .
    .
    .
    300 END
OK *RUN*

Llegados a este punto, pueden haber ocurrido cuatro cosas:

a) Error en la lectura de datos.

b) El programa no funciona correctamente.

c) Error en la escritura de resultados.

d) Todo funciona correctamente.

En cualquiera de los tres primeros casos habrá que revisar el programa y/o los datos, regresando a los pasos de creación de datos y creación del programa. Una vez corregidos los posibles errores, se repetirá el proceso hasta llegar a la verificación del programa.

Esta operación debe hacerse cuidadosamente. Sin duda alguna, la mayor inversión en tiempo, durante el desarrollo de un programa medianamente complejo, tiene lugar en las etapas de corrección y verificación del mismo. La verificación de los resultados exige su visualización en la pantalla. Para ello será preciso volver al sistema operativo:

OK *SYSTEM*
A> *C:PRINT \ DATA \ ORDEN.DAT*

Aunque el programa se haya ejecutado correctamente, ello no garantiza que esté totalmente exento de errores. Es aconsejable ejecutarlo varias veces, evaluando casos extremos para comprobar la plena corrección de su funcionamiento.

- *Documentación*

El último paso, aunque no por ello menos importante, consiste en documentar el programa. Esta documentación debe contener los elementos mínimos que garanticen al usuario una correcta operación con el programa (descripción de los datos de entrada y salida, consideraciones especiales, ubicación, etc.). Debe incluir además una breve descripción de su funcionamiento interno, algoritmos que utiliza y límites de aplicación.

De esta forma, el programa estará siempre dispuesto para su posible explotación y podrá ser modificado posteriormente por otra persona distinta de quien lo confeccionó, sin que ello suponga un esfuerzo excesivo.

Consideraciones finales

El supuesto descrito a través de los párrafos precedentes, intenta dar una visión general de los pasos a seguir para el desarrollo armónico y estructurado de programas. Desde luego, no se trata de una guía que deba seguir a pies juntillas, puesto que en otros casos que no compartan la estructura del indicado como ejemplo, su aplicación puede ser incluso contraproducente. No obstante, su filosofía general puede aplicarse a la mayor parte de las situaciones, ayudando a un mejor mantenimiento y uso de la información.

OASIS

La potencia al alcance de los microordenadores



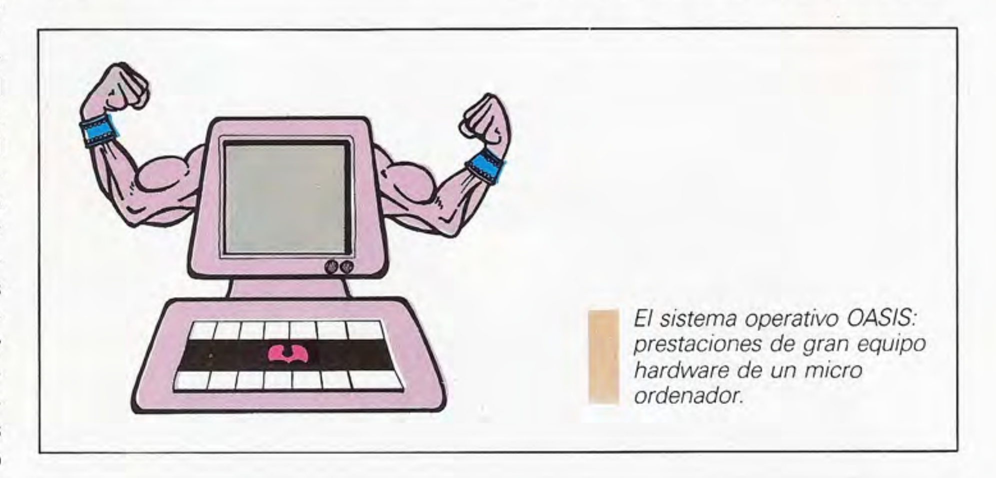
Tal y como se ha señalado al hacer mención a ciertos sistemas operativos, el éxito de los mismos está fundado en su capacidad de respuesta a una determinada solicitud. Así, el éxito del sistema operativo MS/DOS tuvo su base en que respondía a las necesidades establecidas por IBM cuando esta compañía consideró el lanzamiento de su ordenador personal: el IBM-PC. Para considerar el éxito del sistema operativo OASIS es conveniente dar un repaso al campo actual de usuarios de equipos informáticos. Por un lado se encuentran las grandes compañías, usuarios tradicionales de los mismos, normalmente con una estructura de uso plenamente establecida y definida, así como con personal especializado en informática. Por otro lado están los pequeños usuarios de equipos informáticos, muchos de los cuales no han considerado nunca la adquisición de un ordenador y que, potencialmente, dan cuerpo a una gran parte del mercado informático. Estos pequeños usuarios constituyen, como grupo, una mezcolanza heterogénea en cuanto a usos y necesidades; si bien, en su centro destaca básicamente la necesidad de un equipo de elevada flexibilidad y prestaciones aceptables.

Este tipo de usuarios no puede acceder, normalmente por razones monetarias, a equipos informáticos de gran capacidad, como pueden ser minis o superminis. Igualmente, el volumen de información que gestionan no es tan grande que requiera uno de estos equipos, aunque ello no significa que el tipo de gestión que realizan con la información sea distinto al que lleva a cabo una gran empresa. Evidentemente, el tratamiento de una factura en su forma intrínseca es igual en una pequeña tienda que en una gran empresa. Lo expresado indica que, en la mayoría de los casos, el referido segmento de potenciales usuarios se inclinará por la alternativa que supone el microordenador, ya sea de mayor o menor capacidad.

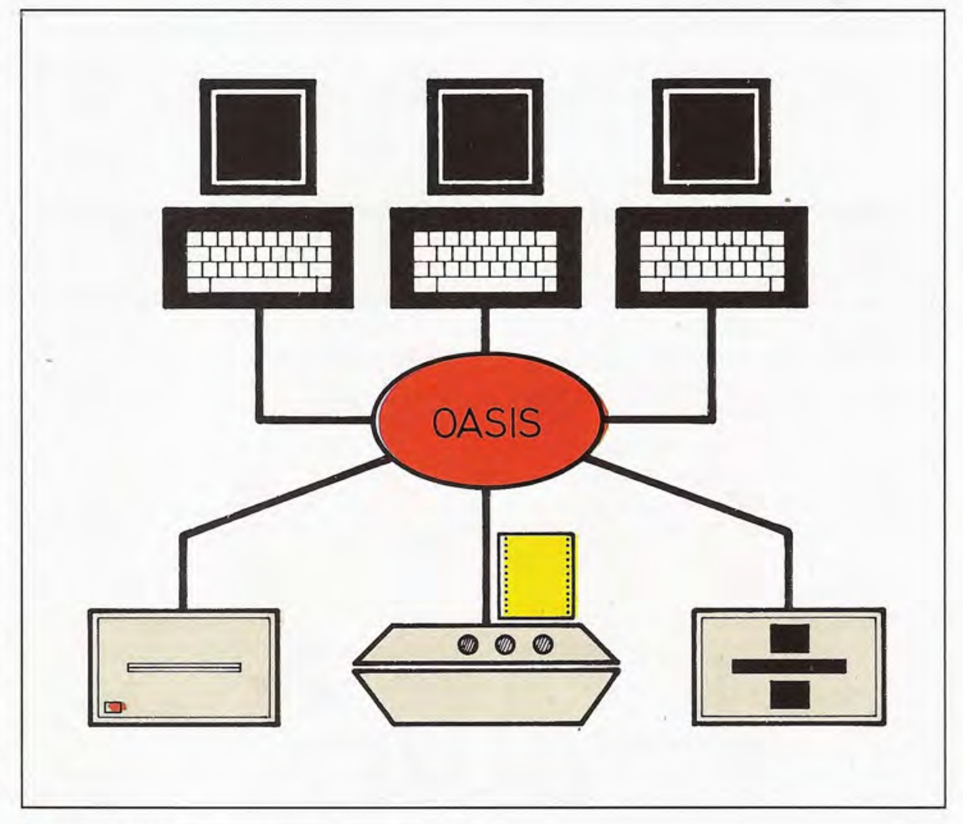
Ahora cabe hacerse la siguiente pregunta: ¿Qué sistema operativo elegir? En muchos casos el entorno ha de ser multiusuario, con lo cual han de estar previstos métodos de seguridad y control normalmente no presentes en ciertos sistemas operativos. En otros casos, será necesario que el equipo sea multitarea, con las servidumbres exigidas por esta modalidad de operación. En cualquier caso, parece lógico que el sistema operativo debe ser independiente del equipo.

Aparece el OASIS

En el año 1977 aparece el sistema operativo OASIS, creado por la firma americana Phase One. Un sistema operativo destinado a microordenadores basados en el microprocesador de 8 bits Z 80 A. Casi de inmediato, este sistema



El sistema operativo OASIS: prestaciones de gran equipo hardware de un micro ordenador.



El OASIS es un sistema operativo que permite la operación simultánea de varios usuarios sobre el ordenador y de distintos periféricos asociados al mismo.



El sistema operativo OASIS está especialmente concebido para equipar a microordenadores orientados a mecanizar actividades de gestión en el marco de la empresa.

operativo empezó a ser considerado entre los usuarios de equipos de esta categoría, ya que su filosofía de diseño, según establecieron sus propios creadores, estaba especificamente adaptado a las necesidades de estos usuarios.

La filosofía de diseño de OASIS se resume en los siguientes puntos:

1. El sistema operativo ha de estar concebido para equipar a microordenadores, aunque ofreciendo características propias de miniordenadores o mainframes.

2. El sistema operativo ha de poder trabajar sobre equipos de distintos fabricantes; esto es: debe ser independiente de la máquina sobre la que opera.

3. Debe estar orientado a usuarios sin experiencia informática previa.

De estos puntos, el que ha resultado más difícil de conseguir de manera consistente ha sido el segundo: la independencia de la máquina. Sin embargo, su consecución representa para el usuario de OASIS una enorme ventaja, ya que puede ejecutar sus programas en distintos ordenadores sin tener que efectuar cambios en los mismos.

También resulta complicada la consecución de un sistema operativo «amigable» para el usuario, algo que no es particularmente común en los sistemas operativos clásicos como el CP/M o el MS/DOS. En el caso del OASIS ello se consigue haciendo consistentes todas las funciones del sistema operativo, dando la posibilidad de activar una orden de ayuda («help») en cada comando, y empleando palabras del vocabulario inglés convencional para todos los comandos.

Parece obvio que este sistema operativo ha sido pensado teniendo en mente a todos aquellos usuarios de pequeños equipos que, sin embargo, requieren prestaciones elevadas.

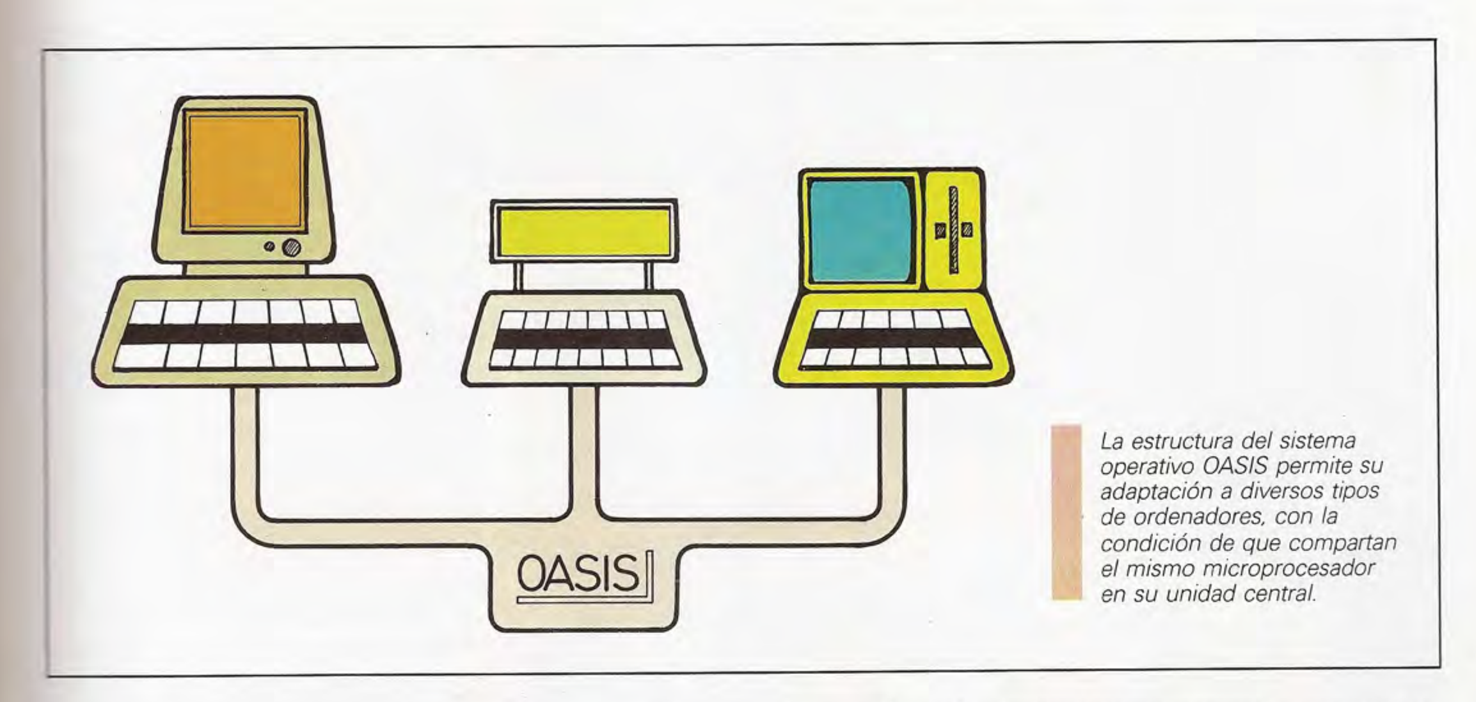


Tal y como señalan sus creadores, el sistema operativo OASIS está orientado a operar en el ámbito de la empresa, controlando el tratamiento de aplicaciones de gestión.

El OASIS desde dentro

Como se ha señalado anteriormente, las primeras versiones de este sistema operativo (hoy existen ya versiones para equipos de 16 bits) estaban preparadas para trabajar sobre microordenadores basados en un microprocesador de 8 bits; permitiendo la operación en régimen multiusuario así como en tiempo compartido.

Sus necesidades, por lo que respecta a memoria residente exigida al ordenador que lo soporta, se concretan en un mínimo de 64 Kbytes; si bien, puede llegar a soportar hasta 784 Kbytes. El núcleo del sistema requiere un total de 16 Kbytes, aunque dependiendo de la versión puede expandirse hasta 32 Kbytes, con lo cual se eliminan alguno de los



La estructura del sistema operativo OASIS permite su adaptación a diversos tipos de ordenadores, con la condición de que compartan el mismo microprocesador en su unidad central.

«overlays» más corrientes, haciéndolos residentes en memoria.

Internamente, el sistema operativo se divide en las tres zonas básicas que se describen a continuación:

• *Núcleo*

Se denomina SYSTEM NUCLEUS. Durante el proceso de carga es lo primero a que se accede, y está compuesto por un conjunto de subrutinas de propósito general que proporcionan y establecen la integración del sistema en un bloque común.

Asociados a este núcleo, aunque físicamente separados, se encuentran los programas que interaccionan con los controladores de periféricos. A través de estos programas el sistema operativo interacciona con los mismos y gestiona las operaciones de entrada/salida, de inicialización de periféricos y de detección y manejo de errores. La concepción de estos programas presenta al usuario diferentes posibilidades, como es el hecho de permitir desconectar un periférico en el caso de que tenga una avería, sin efectos perniciosos para el sistema. O bien cambiar el nombre lógico de un determinado periférico por medio de un comando; por ejemplo, de tal forma que si un programa especifica dirigir una información a la pantalla, ésta puede derivarse hacia la impresora sin necesidad de modificar dicho programa; para ello basta tan sólo con asignar a ésta el nombre lógico dado a la pantalla en el programa.

Una ventaja adicional reside en el hecho de que permite la adopción de periféricos de diferentes fabricantes, al actuar estos programas como si se tratara de verdaderos adaptadores de los periféricos al sistema.

En el caso de que se tratara de la versión multiusuario del sistema operativo OASIS (la más usual y difundida), el núcleo es el encargado de controlar la gestión de memoria así como la compartición de recursos entre los diversos usuarios.

• *CSI*

El CSI es la abreviatura de lo que se denomina «Command String Interpreter». Esta zona actúa como controlador de las operaciones de acceso al sistema y a los programas de usuario.

Una vez lanzado el sistema, la función del CSI es la de buscar en el disco que lo contiene los comandos introducidos por el usuario, pasando el control a los mismos si llega a localizarlos, o bien emitiendo un mensaje de error en el caso contrario.

Similar tarea lleva a cabo cuando el usuario especifica el acceso a un programa: efectúa la búsqueda en las diversas unidades de disco y, de localizarlo, lo carga en memoria y arranca su ejecución.

Esencialmente, la combinación del núcleo, junto con los programas de control de periféricos y el CSI, constituye el eje que soporta todas las operaciones de cálculo y entrada/salida.

• *Programas*

Bajo este apelativo tan simple se ocultan los medios de proceso de información que proporciona el sistema operativo OASIS. Estos programas son esencialmente los comandos del sistema además de los procesadores de lenguajes, como el MACRO Assembler o el BASIC, así como el lenguaje de control de procedimientos EXEC. Este último es una poderosa herramienta que opera a través de los programas y comandos del sistema. Un programa en este lenguaje de control permite detallar toda una serie de tareas que debe llevar a cabo el ordenador, sin necesidad de que el usuario se vea obligado a especificarlas una a una. Normalmente, puede emplearse este procedimiento para ordenar que se ejecuten uno tras otro varios programas, en cuyo caso las variables empleadas por estos pueden almacenarse en un fichero con registros de hasta 512 bytes, incluyendo el fichero hasta 255 registros. A través de programas de este tipo pueden llevarse a cabo tareas como enviar mensajes específicos o generales

A pesar de su indudable potencia operativa, el OASIS se caracteriza por su reducida dificultad de uso, hasta el punto de que resulta adecuado para usuarios con escasa formación informática.

a los restantes usuarios del sistema, en el caso de operar como sistema multiusuario.

Una característica a señalar relacionada con este último punto es el hecho de que si un usuario no puede visualizar un mensaje en un momento determinado, este mensaje se almacena y cuando es posible su visualización se presenta al usuario, borrándose a continuación.

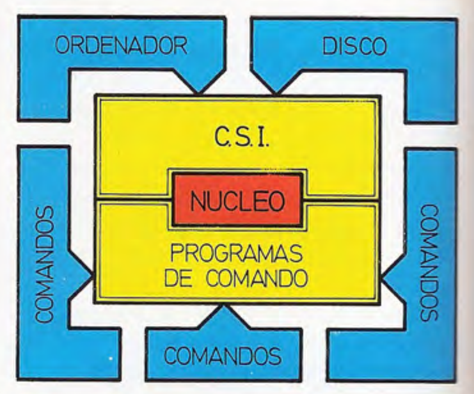
Gestión del sistema

La base esencial de cualquier actividad informática es la gestión de la información de tal forma que ésta resulte fácilmente inteligible para el usuario; en efecto, éste no debe perder un tiempo precioso en descifrar el método por el cual puede acceder a los datos.

Esta ha sido precisamente la directriz esencial adoptada a la hora de desarrollar el sistema operativo OASIS, teniendo en cuenta que su destino es un ambiente de empresa, donde priman las consideraciones de productividad.

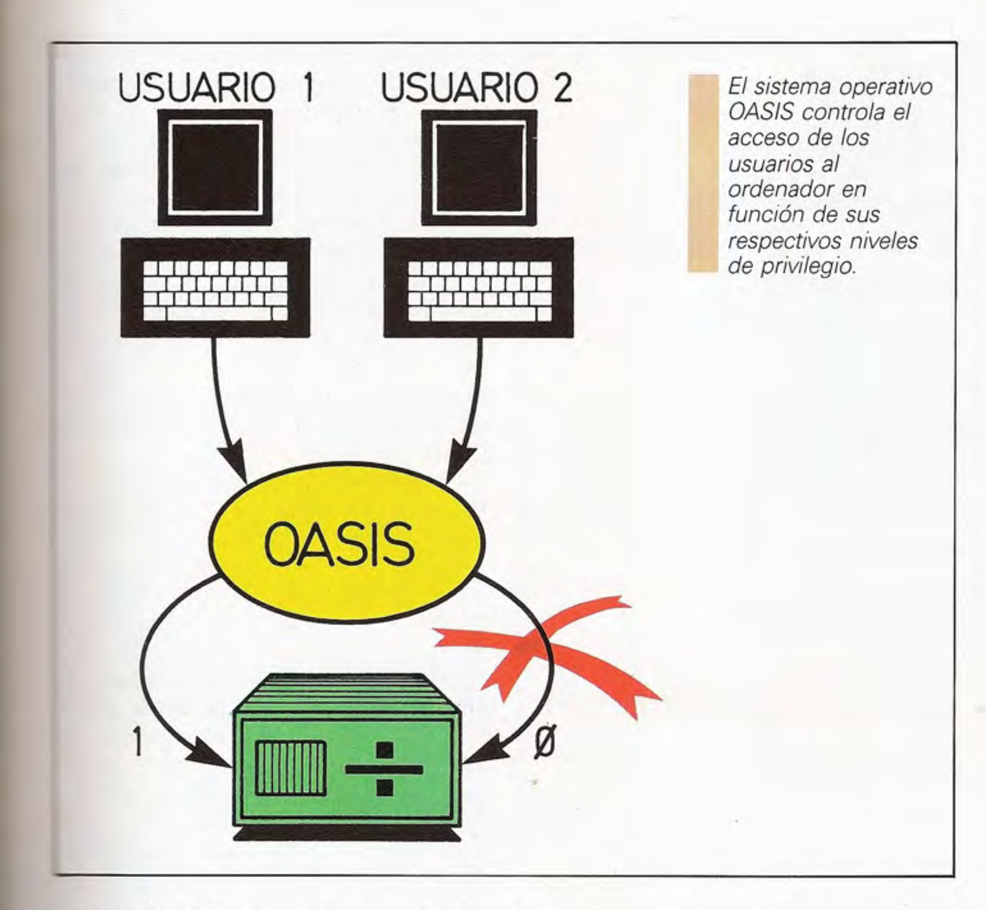
En base a estos criterios de productividad, el sistema operativo OASIS presenta toda una serie de características propias de sistemas más avanzados y caros, que permiten la gestión de la información en un entorno multiusuario; permitiendo o impidiendo el acceso a la misma según los niveles de prioridad otorgados a cada usuario, o bien llevando a cabo un control de las personas que acceden al sistema, a través de registros históricos. Este último factor permite —en el caso de que el ordenador sea utilizado para realizar trabajos a cargo de otra compañía—, presentar a esta última la correspondiente facturación justificada.

Las funciones de gobierno del sistema operativo suponen algo más que controlar el acceso del personal a los recursos informáticos del equipo. También representan el establecer unos niveles de acceso a la información, dependiendo de la categoría o de las necesidades específicas de acceso a la misma para garantizar el buen funcionamiento de la empresa.



Estructura básica del sistema operativo OASIS.

En el sistema operativo OASIS, esta función se lleva a cabo en dos niveles. El primero definiendo una serie de protecciones para los ficheros, de tal forma que cada persona sólo pueda acceder a sus propios ficheros o a otros que se le permitan de forma controlada por el ad-



El sistema operativo OASIS controla el acceso de los usuarios al ordenador en función de sus respectivos niveles de privilegio.

ministrador del sistema. El segundo, definiendo una serie de niveles de privilegio o prioridad (hasta cinco niveles) los cuales se asocian al usuario en el instante en el que éste se conecta al sistema. El nivel de prioridad que tenga otorgado permitirá a cada usuario llevar a cabo o no, una serie de operaciones por medio de los comandos pertinentes. Por lo que respecta al primero de estos métodos, cabe precisar que los ficheros en el sistema operativo OASIS se dividen en tres tipos o categorías: privados, compartidos y del sistema. Los ficheros privados sólo pueden ser accedidos por la persona que los creó y siempre de acuerdo con el nivel de privilegio que tenga asignado.

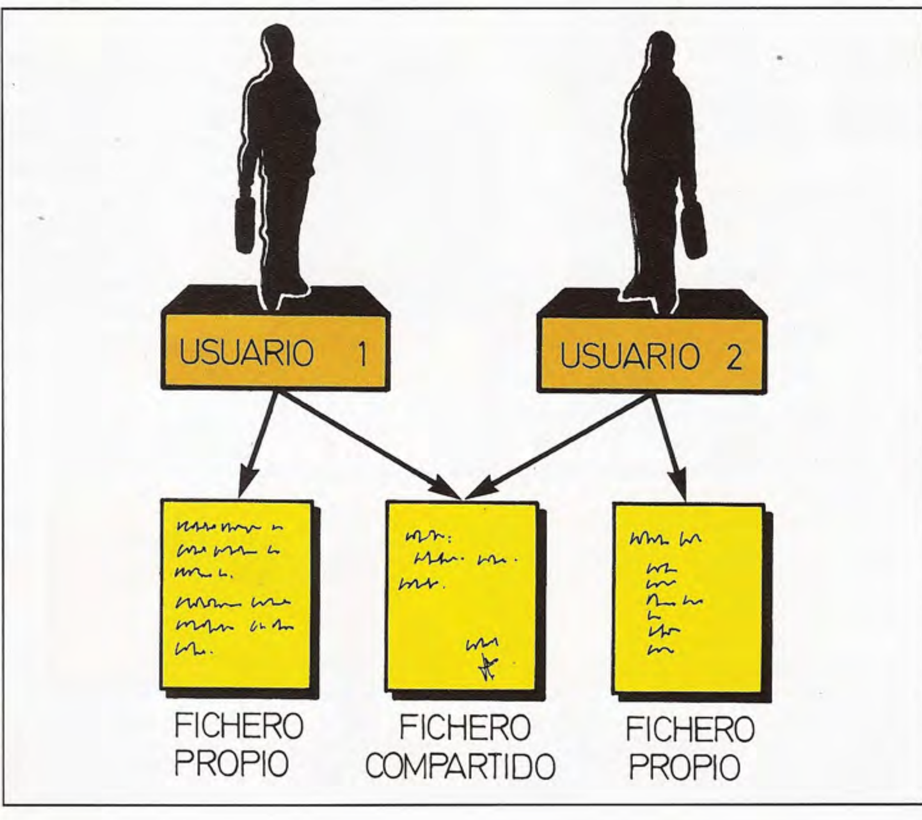
Los ficheros compartidos son creados por un usuario específico; si bien, por necesidad, en función de las características de la información contenida en los mismos, se hace necesario que otras personas tengan acceso a los mismos.

La tercera categoría de ficheros, de acuerdo al criterio establecido, está ocupada por los ficheros del sistema. Estos contienen comandos propios del OASIS y a los cuales acceden los usuarios en función de sus niveles de privilegio.

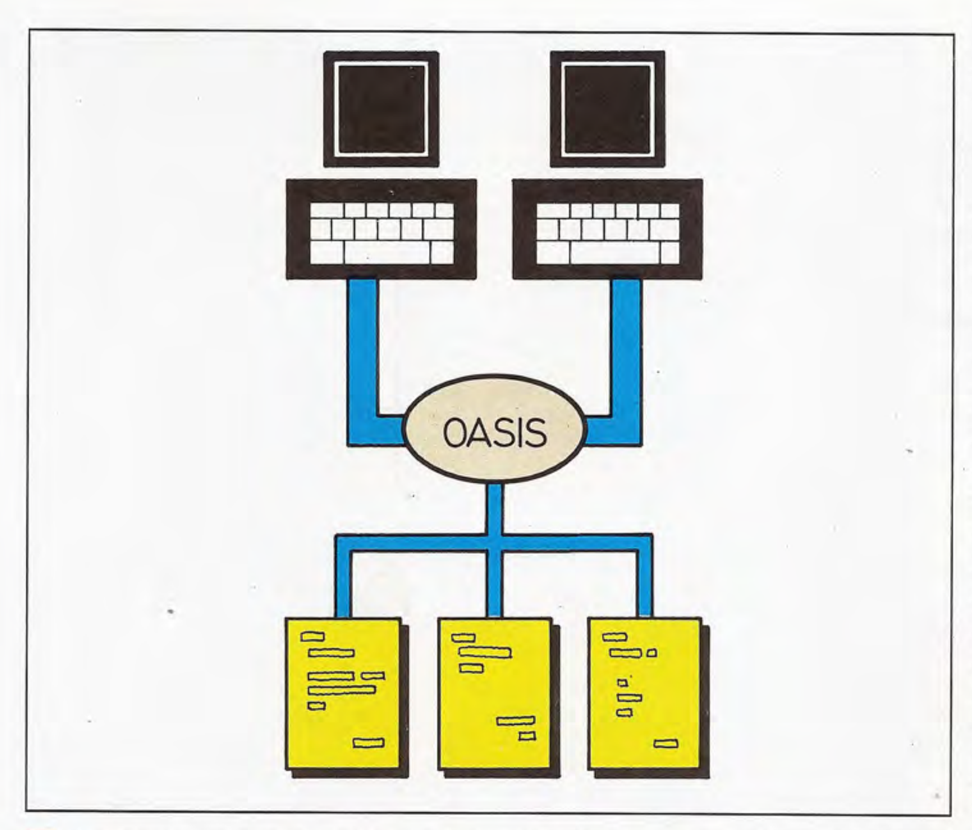
Para ilustrar este principio de funcionamiento es conveniente proponer un ejemplo. Suponga que existen dos usuarios, inicialmente con el mismo nivel de prioridad y cuyos ficheros están dirigidos a dos áreas de actividad distintas: Almacén y Ventas, respectivamente. Aparte se encuentran los ficheros propios del sistema. La denominación de los referidos ficheros es la que sigue:

- Ficheros del sistema

BASIC. COMMAND
RUN. COMMAND
EDIT. COMMAND
ERASE. COMMAND
RENAME. COMMAND
FILELIST. COMMAND



Una de las peculiaridades del OASIS reside en que admite la existencia de ficheros exclusivos de cada usuario y ficheros compartidos entre varios usuarios.



Los ficheros propios del sistema operativo OASIS resultan accesibles a todos los usuarios conectados al equipo informático.

- Ficheros usuario 1 (Ventas)

VENTAS. MADRID
VENTAS. MALAGA
VENTAS. BILBAO

- Ficheros usuario 2 (Almacén)

ALMACEN. GENERAL
ALMACEN. PARCIAL 1
ALMACEN. PARCIAL 2

En un principio, los ficheros que corresponden a cada usuario serán de uso exclusivo para cada uno de llos. No obstante, parece lógico pensar que el usuario 1, a la recepción de un pedido, debe ser capaz de conocer si hay existencias de ese elemento o no; en consecuencia, sería necesario que éste tuviera acceso al fichero del usuario 2 denominado ALMACEN GENERAL. Este fichero sería entonces un fichero compartido por los dos usuarios, mientras que ambos gozarían de idéntico acceso a los ficheros del sistema, puesto que su nivel de privilegio es el mismo. Supongamos ahora que, por cualquier motivo, no es oportuno que el usuario 2 pueda modificar sus ficheros, mientras que el usuario 1 sí ha de poder hacerlo. Ello se consigue con suma facilidad sin más que modificar los niveles de privilegio de forma que se produzca este hecho. Hay que tener en cuenta que los cinco niveles de privilegio tienen una directa relación con el número y tipo de comandos que el usuario tendrá a su alcance, para hacer uso de los mismos.

Asociadas a todo este proceso de control están las palabras clave que gobiernan el acceso al sistema, y el propio fichero histórico del sistema. Este último, cuyo nombre es SYSTEM HISTORY, almacena en su interior los diferentes registros históricos que se producen cuando un usuario se conecta al sistema; dichos registros almacenan la denominación del usuario que se conectó, así como el tiempo que permaneció conectado; además de otra información adicional, y plenamente necesaria, como puede ser la memorización del momento en el que se obtuvo la copia de seguridad de un disco (Backup) o cuando se produjo la carga del sistema. Normalmente, cuando se llene todo el espacio destinado al fichero histórico, residente en el disco del sistema, el ordenador lo indicará a través de un mensaje al efecto para que los usuarios puedan tomar las medidas oportunas.

El OASIS como sistema multiusuario

Se ha mencionado anteriormente que el OASIS puede trabajar como sistema operativo multiusuario, en cuyo caso el núcleo del sistema operativo controla el reloj interno del sistema y asigna tiempo de microprocesador a las diversas tareas a realizar Ello se realiza de forma tal que, bajo la perspectiva del usuario, el sistema aparece totalmente transparente en sus funciones; como si dicho usuario fuera el único que estuviera accediendo a sus recursos.

El proceso seguido para poner en práctica tal fraccionamiento, consiste en



El propio sistema operativo lleva un control permanente de la actividad de los distintos usuarios a través del denominado fichero histórico.

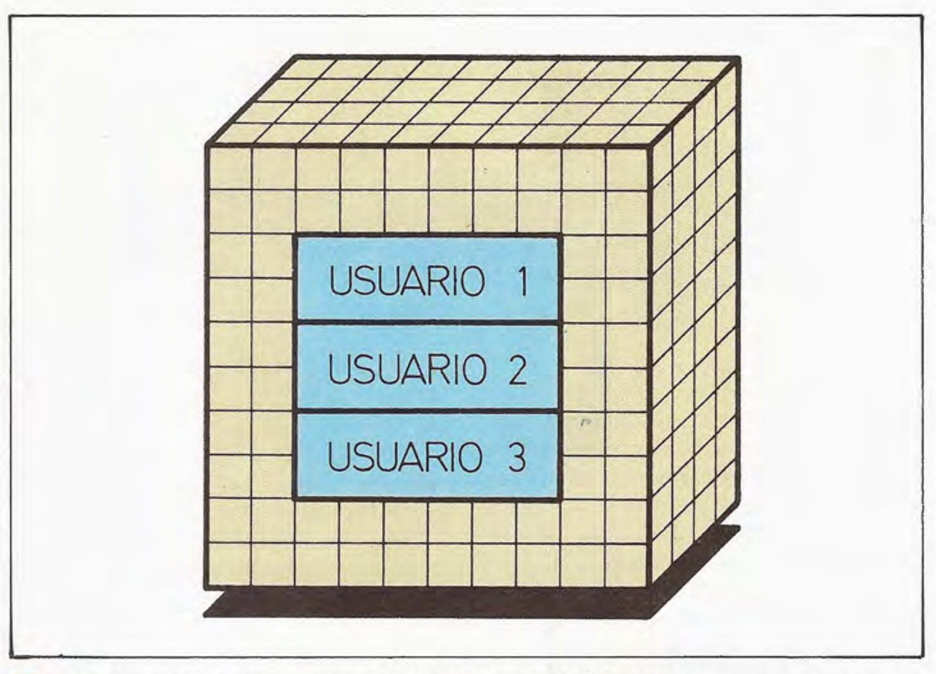
fijar el intervalo de tiempo que se va a destinar a cada tarea, en función de las distintas necesidades de los usuarios. Este cometido lo lleva a cabo el administrador del sistema por medio de la orden SET SLICE. Una vez precisada semejante división, se inicia el trabajo en modo multiusuario, durante el cual cada usuario accede durante la fracción de tiempo establecida a los recursos de la máquina. Transcurrido este tiempo, el control se transfiere al siguiente usuario, siguiendo un proceso rotativo.

La técnica puesta en práctica garantiza un empleo óptimo de la CPU, toda vez que ésta se encuentra en operación de un modo casi constante. Si en un momento determinado algún usuario está cursando una operación de entrada/salida, el intervalo de tiempo de acceso a CPU que le corresponde sería otorgado a otro usuario asociado al sistema. Hay que tener en cuenta que, por sus características, muchos tipos de operaciones de entrada/salida no consumen tiempo de CPU.

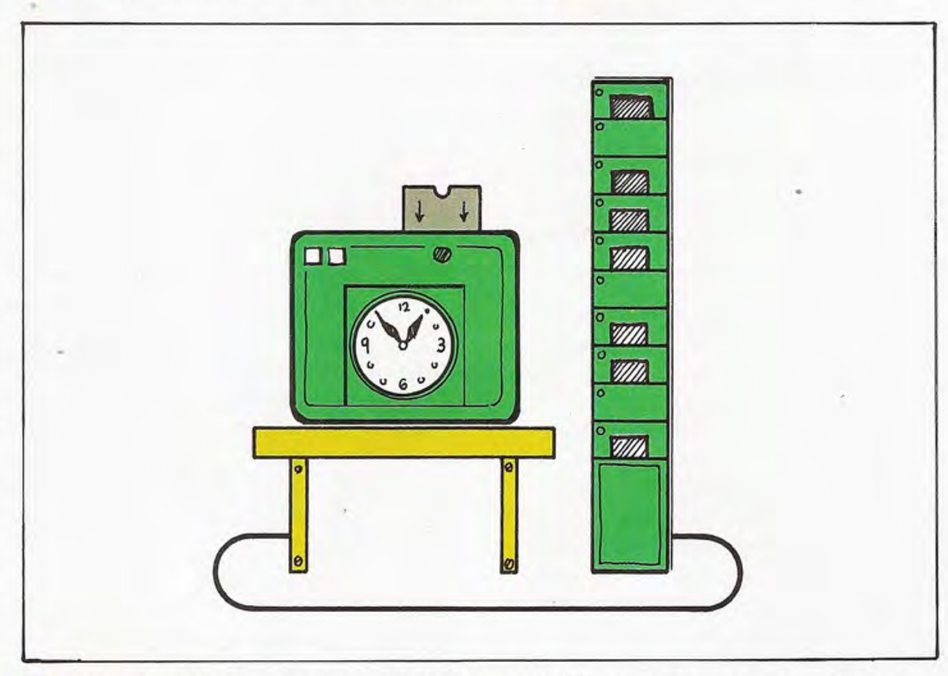
Cuando el ordenador opera en modo multiusuario, su memoria principal se divide en una serie de zonas denominadas «bancos de memoria», las cuales, a su vez, son subdivididas internamente en particiones de memoria. El espacio en bytes asignado a cada una de estas divisiones es variable y, en cualquier caso, definible por el usuario dentro de unos ciertos valores admisibles por el sistema.

Un punto fundamental en la protección de cualquier sistema multiusuario reside en la protección de los registros de los ficheros frente al acceso simultáneo por parte de varios usuarios. En efecto, suponga un hipotético caso en el que dos usuarios, sin conocimiento del hecho debido a la trasferencia del sistema operativo, acceden a la información contenida en un registro, información que supondremos de tipo numérico. Esta información es transportada a sus respectivas particiones de memoria, y procesada, con lo cual se altera su valor y, finalmente, es devuelta a su registro de origen. Si no existiera ningún tipo de protección, el dato que quedaría almacenado sería el correspondiente al usuario que devolviera en último lugar el dato de los dos que lo leyeron simultáneamente.

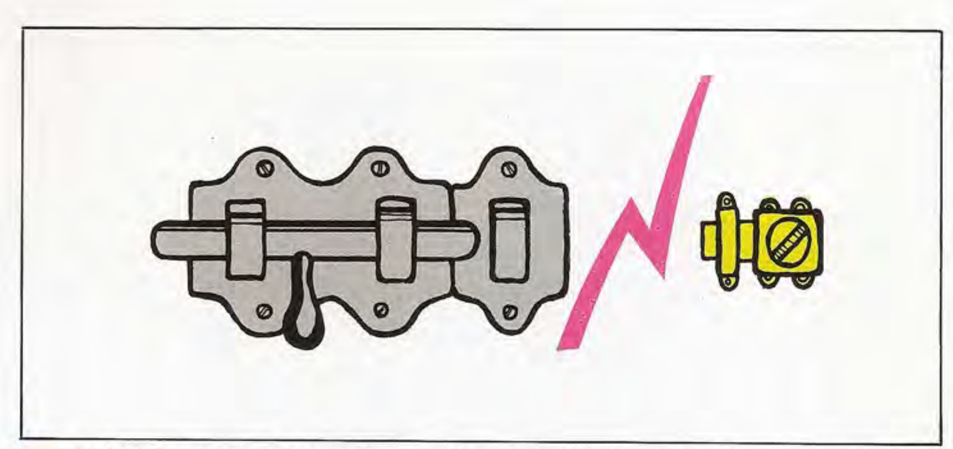
Para obtener una idea de los problemas a que ello puede conducir, cabe imaginar que el dato en cuestión coincide con las existencias de un determinado producto, y las operaciones realizadas por los usuarios son actualizaciones por pedidos. Si el valor inicial del dato es 10 unidades y las operaciones son la sustracción de 8 y 7 unidades por haberse producido pedidos por este valor, puede suceder que el dato actuali-



Los recursos de memoria son divididos por el OASIS entre los diversos usuarios conectados. A su vez, las zonas destinadas a los usuarios son fraccionadas por el sistema operativo en participaciones de tamaño variable.



A través del fichero histórico del sistema, el OASIS mantiene un pleno y detallado control de la actividad de los distintos usuarios con el ordenador.



El OASIS permite definir distintos niveles de protección de los datos, lo cual limita el acceso a la información puesta en juego en el ordenador.



El tratamiento que deben recibir los datos difiere según sea el origen de los mismos. Este es un hecho que debe contemplar el sistema operativo y, en consecuencia, gestionar, según corresponda, los datos residentes en disquetes, disco rígido o cinta magnética.

zado adopte el valor 2 ó 3 unidades, dando la sensación de que todavía hay existencias cuando en realidad es que uno de los dos pedidos no pueden cumplimentarse por falta de elementos.

El sistema operativo OASIS solventa este problema por medio de bloqueos de información de dos tipos: uno a nivel de fichero y otro a nivel de registro. En el primero de los casos, el primer usuario que accede a un fichero se adueña del mismo, y en tanto no finaliza sus operaciones con el mismo ninguna otra persona puede tener acceso. Evidentemente, este procedimiento es interesante en el caso de que se trate de ficheros de escasa frecuencia de acceso, pues de lo contrario produciría retrasos inaceptables a los usuarios. El otro método que brinda el sistema operativo OASIS, es el bloqueo a nivel de registro, más adecuado para ficheros de acceso frecuente. Se trata de un método por el cual el usuario que primero accede a un registro de un fichero provoca un bloqueo del mismo, de tal forma que el siguiente usuario que efectúe el acceso no podrá modificar la información contenida mientras que el anterior usuario no haya finalizado sus operaciones y libere el registro.

Almacenamiento en disco y criterios de nomenclatura

Un sistema operativo es, esencialmente, un surtido de herramientas para gestionar la información. Información que es almacenada y gestionada de una forma variable, dependiendo de las características de los datos, de su estado de proceso, así como de la posible previsión de uso que se va a hacer de los mismos. Resulta obvio que no serán tratados de igual forma los datos residentes en un disco rígido, que los contenidos en un disquete o en una unidad de cinta magnética. Igual de importantes son los métodos empleados para la protección de la información, puesto que en un sistema multiusuario es evidente que existen posibilidades no factibles en un sistema monousuario.

En el sistema operativo OASIS se ha prestado una especial atención a estos aspectos. Especialmente, al hecho de facilitar una gestión de los ficheros lo más transparente posible, para que los usuarios con poca experiencia no se vean desbordados ante complicados comandos de operación con ficheros, tales como los empleados en sistemas opertivos como el CP/M en sus distintas versiones, o el MS/DOS. Por lo demás, el OASIS presenta toda una serie de posibilidades para el bloqueo de la información ante posibles accesos indeseables en un ambiente de operación multiusuario.

El sistema operativo OASIS está configurado de tal forma que admite la operación de hasta ocho unidades de disco, lo que supone una respetable capacidad de almacenamiento. El sistema gestiona cada una de las unidades de disco por separado, con lo cual se inhibe la posibilidad de que un fichero pueda comenzar en un disco y continuar en otro, obligando a que cada fichero esté contenido en un solo disco. Como consideración adicional, cabe indicar que un fichero sólo podrá tener como tamaño máximo el del disco en el que reside.

Todo disco que vaya a ser utilizado con el sistema operativo OASIS ha de ser inicializado previamente; de lo contrario, no será posible su empleo como medio de almacenamiento. Esta operación la lleva a cabo un programa que forma parte del propio sistema operativo y que es invocado a modo de comando. Se trata del comando INITDISK, el cual hace uso en su interior de la opción FORMAT.

Desde luego, un disco puede ser inicializado cuantas veces se desee, aunque hay que tener en cuenta que este proceso elimina la información almacenada en el mismo, la cual no puede ser recuperada posteriormente.

La información que se graba en el dis-

co durante el proceso de inicialización o formateado se divide en tres bloques:

— Directorio
— Etiqueta del disco
— Mapa del disco

La función del directorio es la de actuar a modo de índice o agenda de información para el sistema operativo cuando este accede al disco. En efecto, el directorio contiene información sobre el contenido del disco; esto es, acerca de sus diferentes ficheros, así como los punteros que indican la situación de los ficheros. Resulta evidente que sin esta información el sistema operativo se encontraría impotente a la hora de extraer información de los periféricos de almacenamiento masivo.

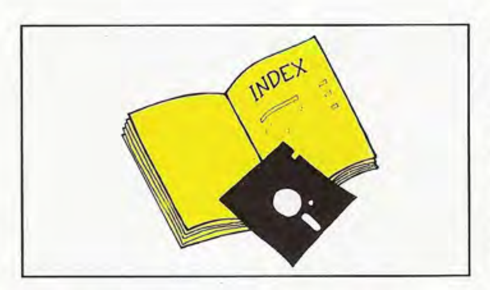
El mapa del disco tiene por objeto llevar un control de las áreas del disco empleadas para el almacenamiento de información; estas zonas se dividen en una serie de bloques, que en el caso del OASIS son de 1 K, es decir, 1.024 bytes de información.

Criterios de búsqueda de ficheros

Cuando el usuario de un ordenador equipado con el sistema operativo OASIS comunica su deseo de acceder a un fichero determinado, ha de especificar los siguientes datos:

— Nombre del fichero
— Tipo de fichero
— Disco en que está almacenado

Del nombre del fichero y de su tipo hablaremos posteriormente. Vamos ahora a ocuparnos del disco. Este puede ser identificado de dos formas. La primera de ellas especificando la etiqueta del disco, la cual, como se recordará, fue generada por medio del comando INITDISK. La segunda, a través de una etiqueta de directorio de un carácter, la cual se establece por medio del comando ATTACH. La función de este comando es crear una asociación entre un dispositivo físico y un dispositivo lógico, permitiendo así que el operador se adapte a los periféricos de que dispone, o cambie de forma dinámica el periférico de entrada/salida sin necesidad de efectuar ninguna modificación en los programas que está manejando. Como



La función del directorio es la de actuar a modo de índice o agenda de información que facilite la tarea del sistema operativo cuando éste debe acceder al disco.

se ha mencionado, el sistema operativo OASIS puede controlar hasta ocho unidades de disco. Estas unidades reciben una denominación específica que es:

S, A, B, C, D, E, F, G

donde S es el disco del sistema operativo. La letra que se designa a la unidad de disco en la que se encuentra almacenado el fichero al que se desea acceder, es la etiqueta de directorio a especificar en el caso de no haber indicado la etiqueta del disco.

Puede darse el caso de que, por cualquier motivo, no sea posible especificar el disco en el cual se encuentra almacenado un determinado fichero. En tal situación, el sistema operativo OASIS inicia un proceso de rastreo a través de las diversas unidades de almacenamiento masivo. Este proceso es diferente según lo que el sistema esté buscando. Si se trata de un programa de usuario, los discos son rastreados, uno a uno, y por el orden alfabético de sus referencias, es decir:

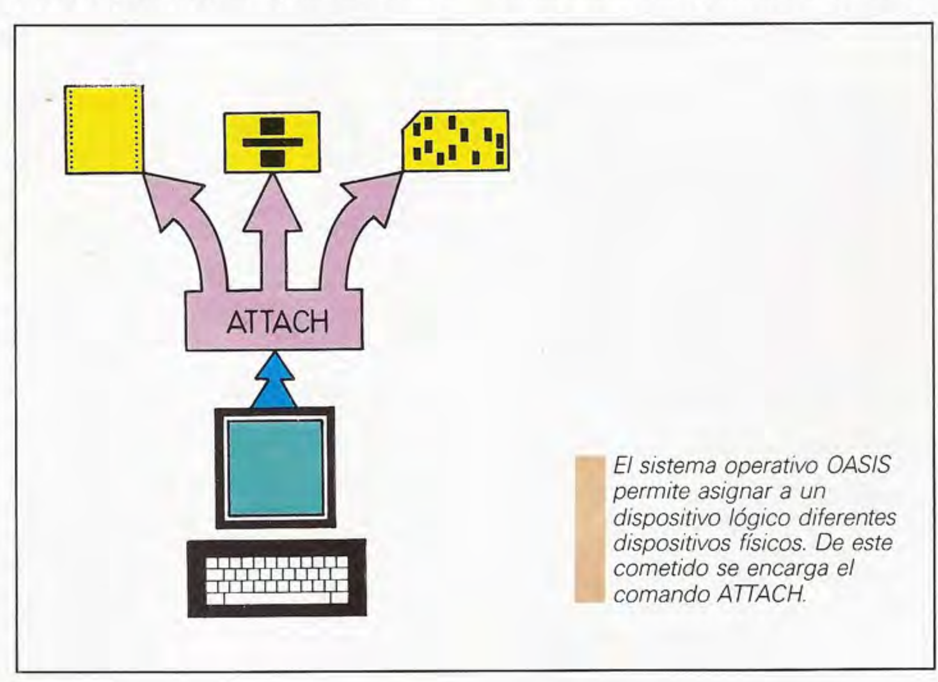
A→,B→,C→,D→,E→,F→,G→,S

Sin embargo, si lo que se especifica es un comando, que por su definición en el sistema operativo deba estar almacenado en disco, éste será buscado inicialmente en el disco del sistema (S) y a continuación en los restantes discos, siguiendo el orden alfabético.

Igual procedimiento se emplearía con un programa EXEC. Como ya sabemos, este es el lenguaje de comandos propios del OASIS, a través del cual pueden encadenarse una serie de operaciones definidas por medio de distintos comandos del sistema.

Criterios de denominación de ficheros

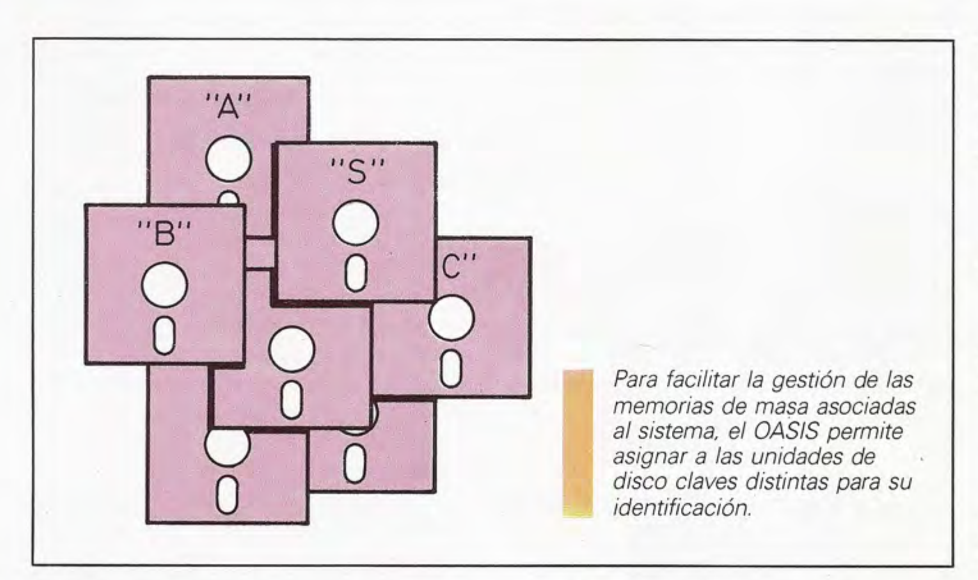
Cualquier fichero que se cree en el transcurso de las operaciones con el or-



El sistema operativo OASIS permite asignar a un dispositivo lógico diferentes dispositivos físicos. De este cometido se encarga el comando ATTACH.



El OASIS presta un especial cuidado a la protección y al bloqueo de los datos frente a posibles accesos no deseables. Hay que tener en cuenta que su ámbito de explotación suele coincidir con un entorno multiusuario.



Para facilitar la gestión de las memorias de masa asociadas al sistema, el OASIS permite asignar a las unidades de disco claves distintas para su identificación.

denador tiene únicamente dos caminos de generación: el propio sistema operativo del equipo, o el programa de usuario que esté trabajando en la actualidad sobre el ordenador. En cualquier caso, el fichero ha de ser denominado de forma tal que se excluya la confusión con otros ficheros. De ahí que sea necesario seguir unos criterios de nomenclatura, similares en la mayoría de los sistemas opertivos, aunque con peculiaridades propias en cada uno. En el caso del sistema operativo OASIS, la denominación de un fichero consta de tres elementos:

NOMBRE del fichero. *Tipo* del fichero. *Disco* en el que se almacena

El significado de cada uno de estos elementos es

— Nombre del fichero: incluye de uno a ocho caracteres de longitud, siendo el primero de estos caracteres obligatoriamente una letra, mientras que los restantes pueden contener otras letras, números o el signo «$», no admitiéndose otro tipo de caracteres en esta nomenclatura.

— Tipo de fichero: también debe incluir de uno a ocho caracteres y se utiliza como un caracterizador del nombre del fichero. Sin embargo, hay que tener en cuenta que algunos de estos tipos de fichero tienen un significado particular para el sistema operativo OASIS y, por lo tanto, han de ser empleados en consecuencia. Por lo que se refiere al tipo de caracteres, se siguen en este caso las mismas reglas que las aplicadas al nombre del fichero.

— Disco en el que está almacenado. Esto se indica por medio de uno cualquiera de los dos métodos expuestos al hablar de la localización de un fichero; bien por medio de la etiqueta del directorio, o bien por medio de la etiqueta del disco.

En la mayoría de los casos no será preciso especificar este último elemento de identificación, ya que, como se ha visto, el sistema operativo realizará una búsqueda secuencial por los diferentes discos conectados al ordenador.

Formatos de los ficheros

El sistema operativo OASIS emplea seis formatos distintos de ficheros; estos son:

- Ficheros secuenciales ASCII

Estos son ficheros conformes al American National Standard Code for Information Interchange; en ellos, cada carácter está representado por un código de 8 bits. Los ficheros de este tipo son los creados por medio del Editor, programas del MACRO ensamblador, o ficheros de datos creados por el usuario. Este tipo de ficheros son los únicos que soportan registros de longitud variable; aunque, por su carácter secuencial, no

pueden ser actualizados registro a registro, sino que han de añadirse los nuevos registros al final del fichero.

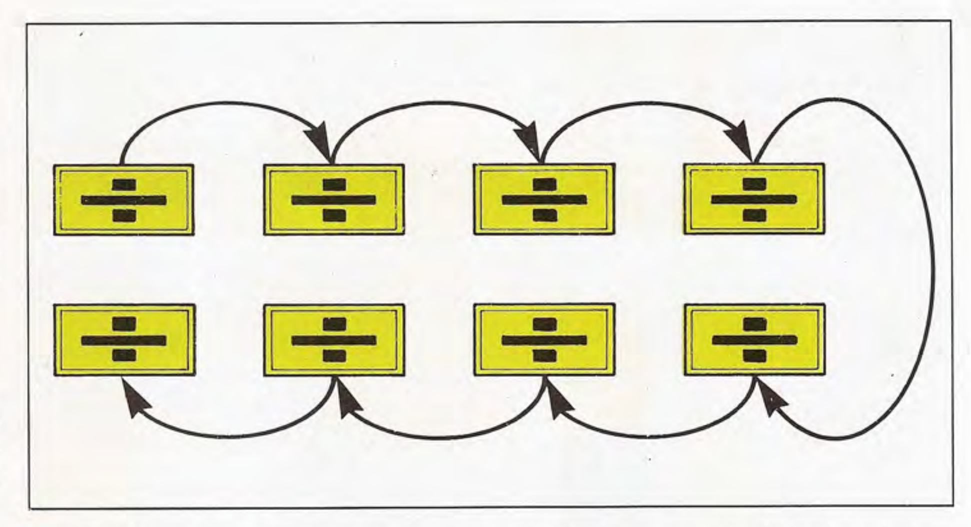
• Ficheros de acceso directo
Contienen datos binarios sobre registros de longitud fija y con un número determinado de registros. El acceso a cada uno de los registros se efectúa especificando un número coincidente con el identificador de dicho registro.

• Ficheros indexados
El sistema operativo OASIS soporta un tipo de fichero de una sola clave, siendo, al igual que los ficheros de acceso directo, de longitud fija y con un número de registros predeterminado. La clave es un campo del registro que actúa como identificador del mismo y que permite acceder a él con independencia de los demás. El OASIS mantiene siempre este tipo de ficheros ordenados por clave.

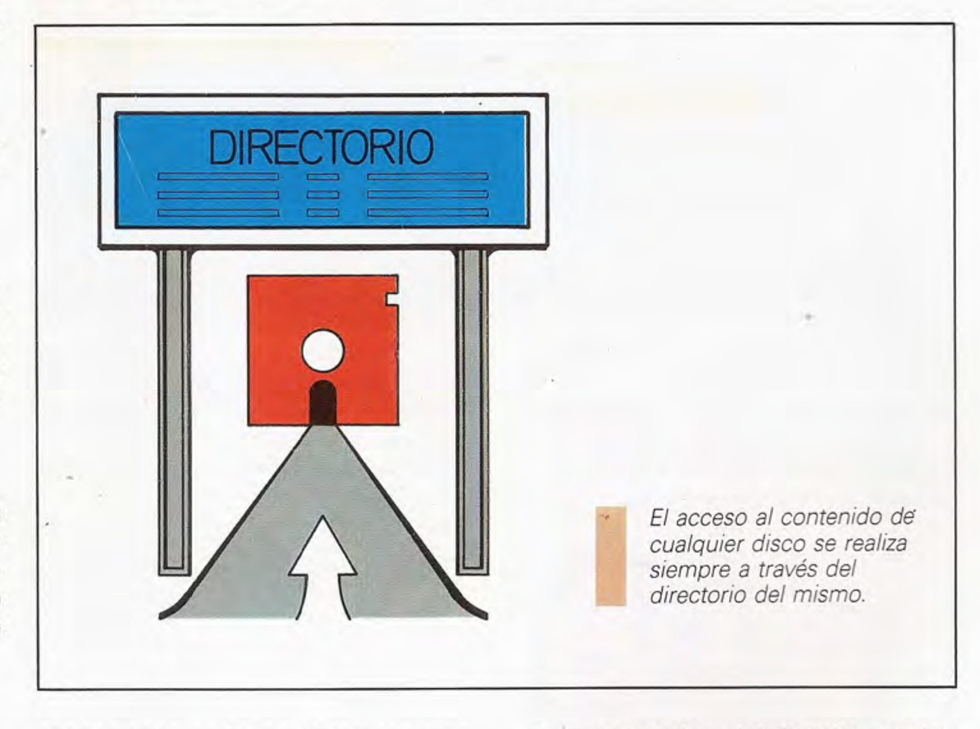
• Ficheros en formato de clave
Son idénticos en su formato a los ficheros indexados, sólo que no se mantienen ordenados por clave.

• Ficheros en formato absoluto
Son ficheros generados por el programa LINK-Editor y están restringidos a los programas en código máquina. Contienen a los programas en un formato idéntico al que tendrán en la memoria del ordenador una vez cargados.

• Fichero en formato reubicable
Son ficheros también reservados a los programas en código máquina, pero en este caso son originados con una dirección de memoria cero, siendo reubicado el programa en memoria a medida que es cargado.



En el caso de que no se especifique la unidad en la que reside el fichero con el que hay que operar, el sistema operativo inicia una búsqueda secuencial a través de los discos alojados en las distintas unidades.



El acceso al contenido de cualquier disco se realiza siempre a través del directorio del mismo.

El vocabulario de comandos

Para desarrollar su trabajo sobre el ordenador, cualquier sistema operativo se apoya en la comunicación entre los periféricos y la CPU, comunicación que el sistema gestiona respondiendo a los dictados del usuario. Estos dictados se confeccionan explotando los comandos disponibles, que a través de sus diversas opciones, permiten definir con exactitud la tarea a realizar.

El sistema operativo OASIS, como ya se ha indicado anteriormente, está esencialmente orientado a usuarios con escasa formación informática; sus comandos se designan empleando, siempre que es posible, palabras del vocabulario inglés convencional. Ello convierte al OASIS en un sistema operativo más «cómodo» para el usuario que otros de similar categoría.

Los comandos del OASIS pueden agruparse en distintas categorías, teniendo en cuenta su función:

— Comandos relacionados con la gestión y ejecución de programas.

— Comandos destinados al control de los parámetros del sistema.

— Comandos destinados al control de las comunicaciones.

— Comandos relacionados con la gestión de ficheros

— Comandos relacionados con la

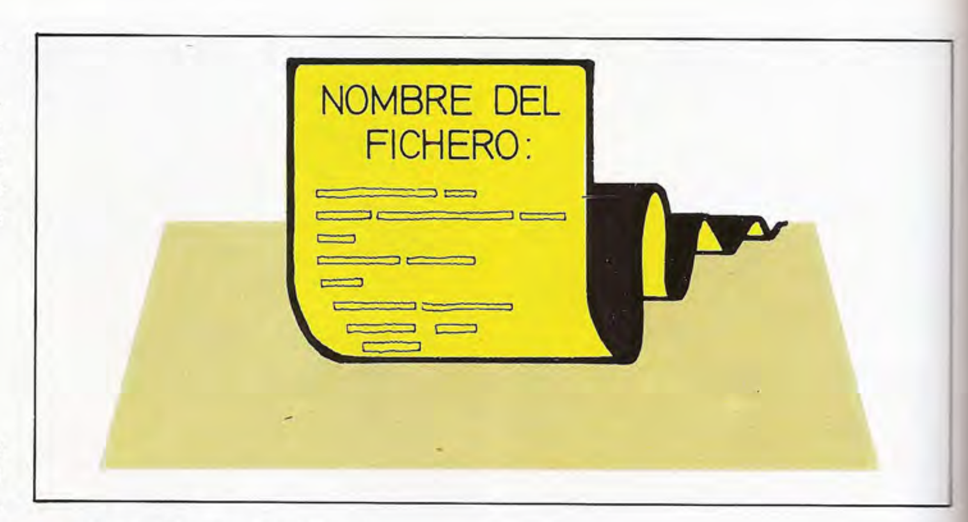
gestión y el control de periféricos de entrada/salida.

Dentro de cada una de estas categorías caben diversos comandos. Dado su elevado número, no vamos a entrar en detalles respecto a todos ellos, sino que tan sólo se dará una idea de aquellos que resultan más representativos.

Comandos para la gestión y ejecución de programas

Dentro de los comandos relacionados con la gestión de programas, tal vez el más representativo sea el editor del sistema. A través del mismo tiene lugar la creación y el mantenimiento de ficheros que serán empleados por programas tales como el procesador de lenguaje EXEC, el MACRO-ensamblador o el intérprete/compilador BASIC. Además, también puede ser utilizado para generar y mantener ficheros destinados a operar con los programas del usuario, tales como ficheros de datos.

El editor es invocado por medio del comando TEXTEDIT, el cual abre al usuario el acceso a un repertorio de comandos de edición. Estos, en su formato y nomenclatura, son congruentes con los comandos generales del sistema. El referido editor es del tipo «de líneas»; ello significa que actúa sobre líneas completas a la hora de ejecutar sus funciones, y no sobre un conjunto de líneas como haría un editor de pantalla completa.



El nombre de un fichero ha de aportar una plena identificación del mismo.

El formato del comando que da entrada al editor es el siguiente:

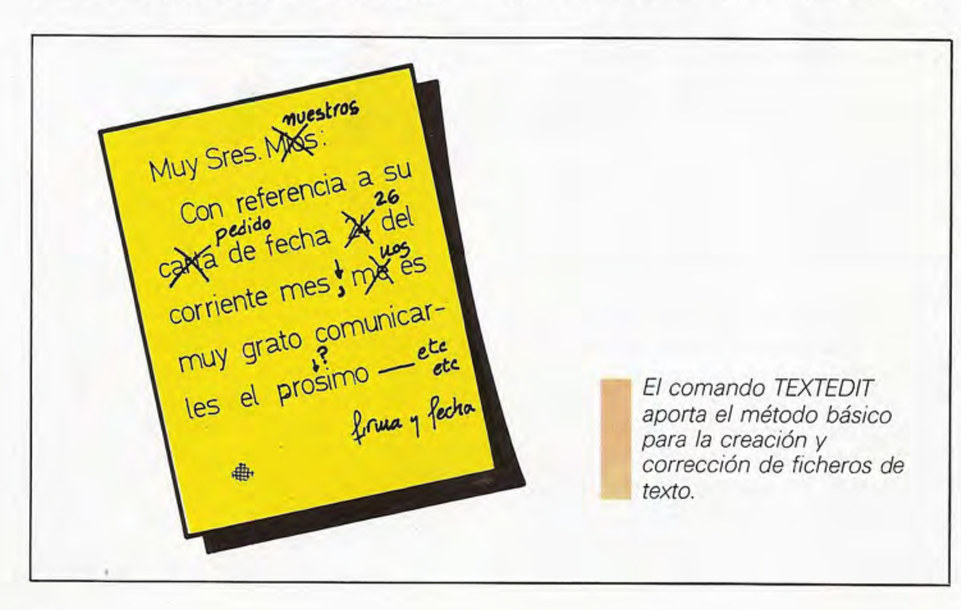
TEXTEDIT (Nombre del fichero)

En estas condiciones el OASIS inicia la búsqueda del fichero especificado. Si en el nombre del fichero no se incluye una referencia al disco en el que se encuentra, el editor realizará una secuencia de búsqueda a través de las diferentes unidades de disco asociadas al ordenador, de acuerdo con lo indicado en otros capítulos de esta obra. Una vez localizado el fichero, éste es cargado en memoria. Por medio de los mensajes oportunos, el sistema comunicará si el fichero ha sido cargado correctamente o si, por el contrario, el tamaño del fichero es superior al de la partición de memoria a disposición del usuario. El «prompt» o indicativo de que el editor está en activo es, en el sistema operativo OASIS y al igual que en el sistema MS/DOS, un asterisco (*). El asterisco, sin embargo, puede desaparecer toda vez que el editor del sistema opeativo OASIS puede operar en dos modos distintos:

- *Modo comando* en el que el prompt se encuentra presente y en el que se introducen los diferentes comandos de actuación sobre las líneas del fichero, como por ejemplo el de borrado.

- *Modo de introducción de texto,* en el cual no aparece el asterisco y se realiza todo el proceso de modificación interna de las diferentes líneas.

El final de la operación con el editor se ordena por medio de dos comandos: FILE o QUIT. El primero de ellos produce la actualización del fichero original, toda vez que éste no ha sido modificado sobre el periférico de almacenamiento masivo, sino únicamente sobre la copia almacenada en la memoria principal del sistema. En todo caso, la versión antigua del fichero no se pierde; el sistema operativo OASIS al igual que hacen otros sistemas operativos como el MS/DOS, almacena esta versión con el mismo nombre y con el tipo de fichero



El comando TEXTEDIT aporta el método básico para la creación y corrección de ficheros de texto.

BACKUP, eliminando durante el proceso cualquier otro fichero con esta denominación. El segundo comando de salida, QUIT, se utilizará cuando el usuario decida abandonar una sesión de trabajo sin almacenar en memoria los datos o modificaciones introducidas; en tal caso, el control se devuelve al entorno en el que se invocó el comando TEXTEDIT.

Un comando que se encuentra íntimamente relacionado con la ejecución de programas es el denominado FORCE. Sólo puede utilizarse cuando el OASIS opera en modo multiusuario, y su misión es la de «forzar» a otro usuario a ejecutar una determinada tarea. El formato de esta orden es el que se indica a continuación:

FORCE (Id. participación) (Comando)

El identificador de participación señala cuál es el número de la participación del usuario o el nombre de la «cuenta» que va a ser forzada a realizar una tarea, mientras que el argumento comando define precisamente la tarea a ejecutar por el otro usuario. FORCE es un comando sumamente práctico, puesto que brinda la posibilidad de acceder a otros usuarios desde el terminal propio, permitiendo, en el caso de que no sea posible efectuar una operación desde el propio terminal, que ésta se realice sobre un terminal ajeno. Aunque, desde luego, su uso es peligroso si no se avisa previamente al otro usuario, ya que puede originar pérdidas de información por implicar una reconfiguración de la memoria.

Comandos para la gestión de los parámetros del sistema

Dentro de este grupo caben distintos comandos cuya influencia en la gestión del sistema es variable. Veamos algunos de los más importantes.

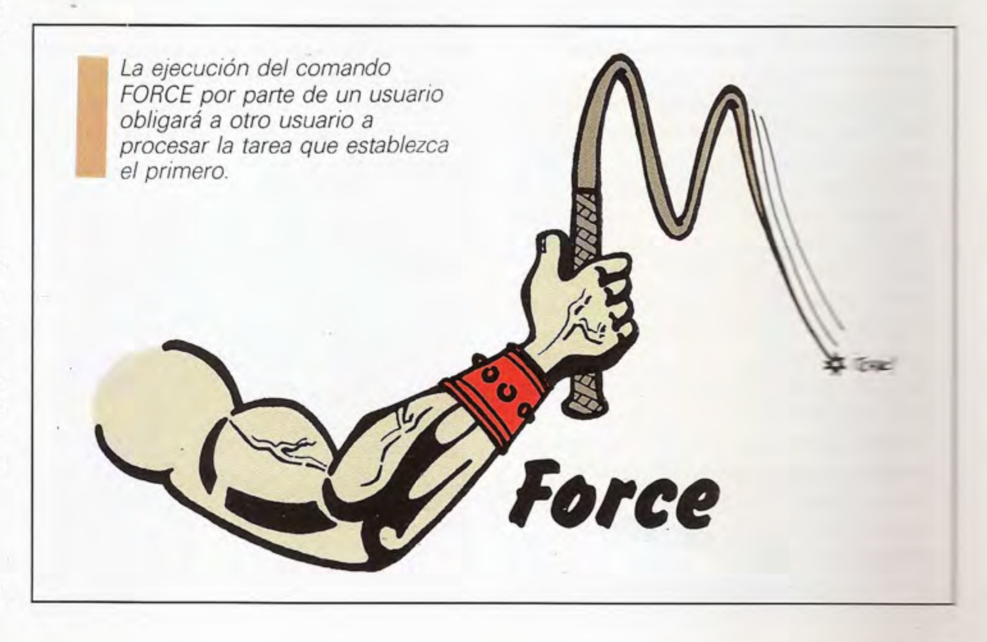
El comando SHARE tiene la función de señalar al sistema operativo que un determinado fichero va a ser compartido por varios usuarios. Su formato es el siguiente:

SHARE (nombre del fichero) (Cuentas)

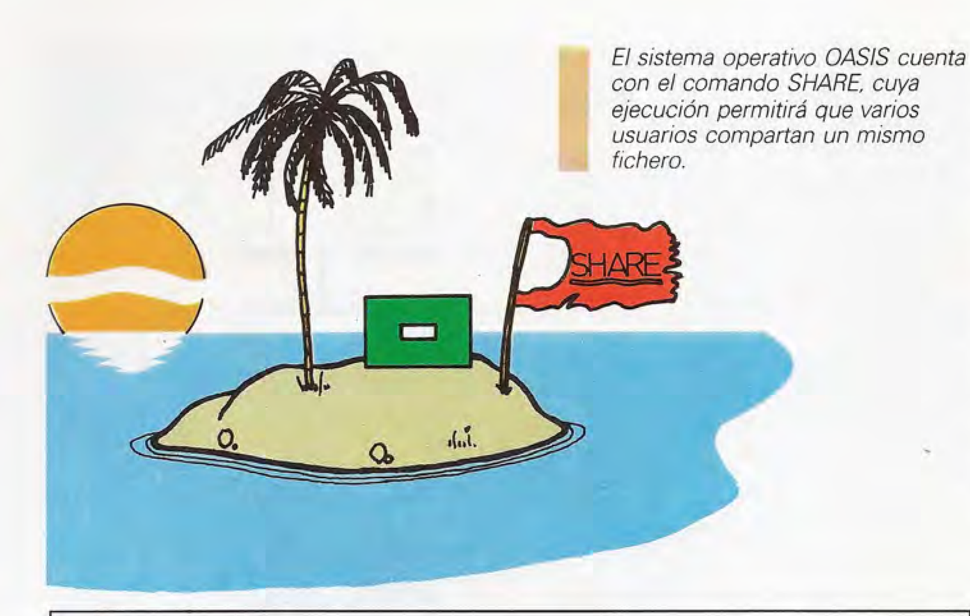
El nombre del fichero corresponde al



El sistema operativo OASIS contempla la necesidad de que los diversos usuarios conectados al ordenador compartan con eficacia los recursos del sistema; por ejemplo, la impresora.



La ejecución del comando FORCE por parte de un usuario obligará a otro usuario a procesar la tarea que establezca el primero.



El sistema operativo OASIS cuenta con el comando SHARE, cuya ejecución permitirá que varios usuarios compartan un mismo fichero.

La protección del software

En los últimos años el campo informático se ha ido abriendo paulatinamente a más usuarios; en gran medida, a través de la senda aportada por los microprocesadores. Cada vez es más amplia la circulación de software y son más y más las personas que tienen acceso a la información. Una de las facetas del software de indudable importancia, es la relativa a la integridad e intimidad de la información. Afrontar este problema lleva inevitablemente a situaciones de difícil resolución. Por una parte, el objetivo primordial de la informática es ofrecer información con la mayor sencillez posible para su fácil entendimiento; mientras que por otra hay que mantener una cierta integridad de la información, para que sea veraz, y además hay que controlar el acceso a la misma: de poco valdría tener una gran cantidad de datos si de ellos no se extrayeran conclusiones correctas.

A la hora de acometer la resolución de este difícil tema, cabe empezar distinguiendo entre dos tipos de información: la información de tipo general que ha de ser de fácil acceso y que va dirigida a cualquier usuario; y la información confidencial, destinada a personas cuyo cometido las obligue a tomar decisiones importantes o comprometidas. Esta última debe quedar al amparo de miradas curiosas, dada su vital importancia y, por lo tanto, debe estar arropada con todas las posibles medidas de protección.

Dentro de estas medidas de protección se encuentran las llamadas palabras clave o «passwords» que impiden el acceso a la información a aquellas personas que las desconocen. Actúan, realmente, como una especie de llave de software, aunque su fiabilidad no es muy alta puesto que no es muy difícil su divulgación entre personal no autorizado.

La codificación de la información también se utiliza para estos fines. En tal caso, la técnica a aplicar se basa en tablas de sustitución de unos caracteres por otros; con ello el acceso a la información cifrada resulta más difícil al ser más complejo el método de enmascaramiento.

Aumentando la dificultad de acceso a la información, cabe hablar del cifrado que se fundamenta en la mezcla de la información, sometiendo el texto a una serie de operaciones matemáticas previstas en un algoritmo de naturaleza pseudoaleatoria; la operación de descifrado resulta ahora prácticamente imposible si no se deduce el algoritmo correspondiente.

Otro método frecuente es el acceso con derechos restringidos a los ficheros que almacenan información, de tal manera que no todos los usuarios puedan leer o escribir en un determinado fichero.

Aun aplicando toda esta serie de medidas de protección, nunca es posible garantizar una plena y total integridad de la información. Siempre existe la técnica opuesta a la utilizada; en todo caso, cuantas más barreras de protección se introduzcan para que los datos no sean accesibles indiscriminadamente, mayor será el nivel de seguridad logrado.



que va a ser compartido entre varias «cuentas» o usuarios definidos en el sistema. El parámetro cuentas, si es indicado, presenta una lista de las diversas cuentas que tienen acceso al referido fichero. De omitir tal opción, permitirá que las cuentas que tienen acceso al fichero sean modificadas.

El comando ACCOUNT tiene por misión controlar el fichero SYSTEM ACCOUNT, el cual lleva el control de la «contabilidad interna del sistema». El referido fichero memoriza el nombre de las cuentas, sus sinónimos, las palabras clave de acceso y los diferentes niveles de privilegio. El formato del comando ACCOUNT es el que sigue

ACCOUNT (Opciones)

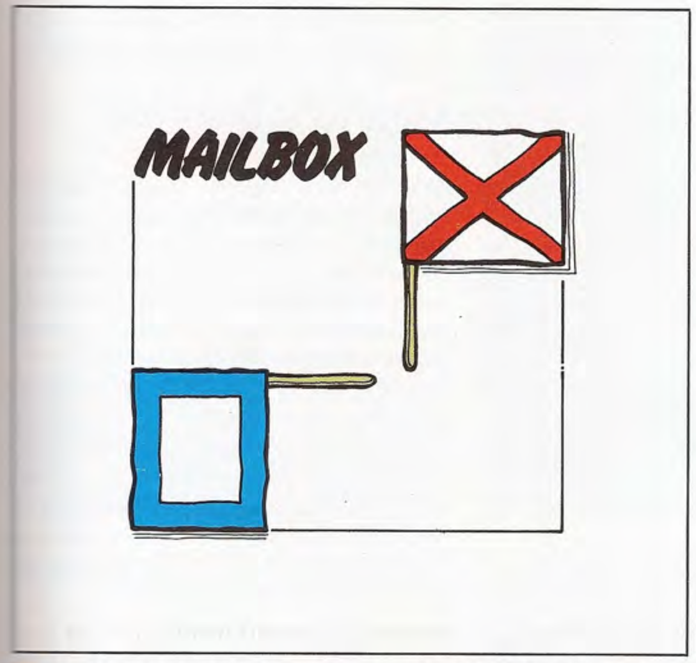
Las distintas opciones que pueden especificarse permiten eliminar usuarios o acceder al fichero histórico en el cual se encuentra almacenada la contabilidad del sistema.

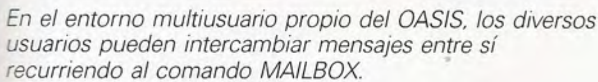
Comandos destinados al control de las comunicaciones

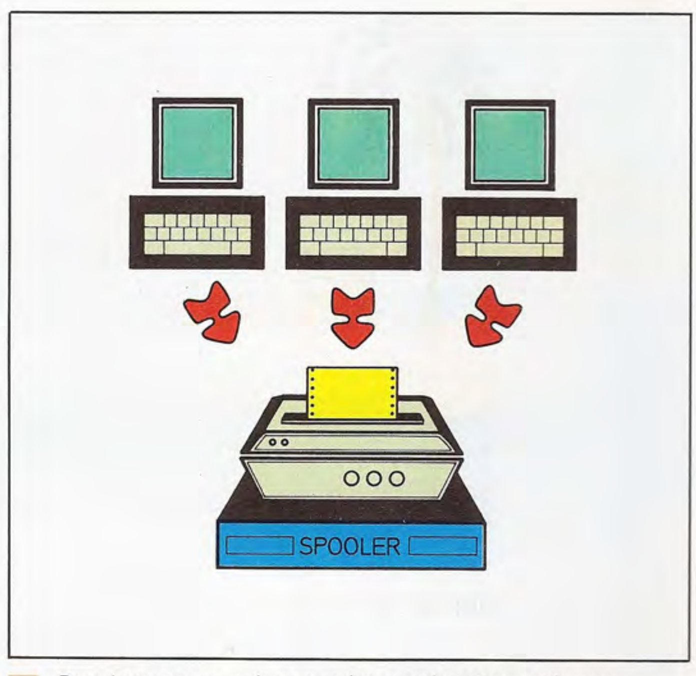
Dentro de este grupo existen varios comandos entre los que cabe citar, por ejemplo, a MAILBOX. Este permite el envío de mensajes entre los diferentes usuarios; desde luego, ello revela que tan sólo se encuentra disponible en la versión mulitusuario del OASIS. Una vez invocado el comando y enviados los mensajes, el sistema operativo mantiene un control de los mismos de tal forma que éstos no se repitan. El sistema operativo garantiza también que cada usuario sólo reciba aquellos mensajes específicamente detinados al mismo.

Comandos relacionados con la gestión de ficheros

El sistema operativo OASIS ofrece una amplia variedad de comandos destinados a la gestión de ficheros; comandos que permiten cambiar el nombre a un fichero, eliminarlo de la memoria, crearlo o copiar un fichero en otros. No




En el entorno multiusuario propio del OASIS, los diversos usuarios pueden intercambiar mensajes entre sí recurriendo al comando MAILBOX.



Para lograr que varios usuarios puedan compartir una misma impresora es preciso hacer uso del «spool» de impresión; el control de esta herramienta corre a cargo del comando SPOOLER.

vamos a profundizar en los habituales, sino que tan sólo citaremos a uno de ellos, peculiar y propio del OASIS. Se trata del comando MOUNT, cuya función es la de informar al sistema que un disco va a ser cambiado. Su formato es:

MOUNT (Ident)*

La zona «ident» corresponde a la etiqueta del volumen o soporte que va a ser cambiado, mientras que el asterisco (*) indica que todos los discos asociados pueden ser objeto de cambio. De producirse un cambio de disco durante un proceso de actualización, sin que el usuario haya invocado a este comando, se obtendrá un mensaje de error.

Comandos para el control de periféricos de entrada/salida

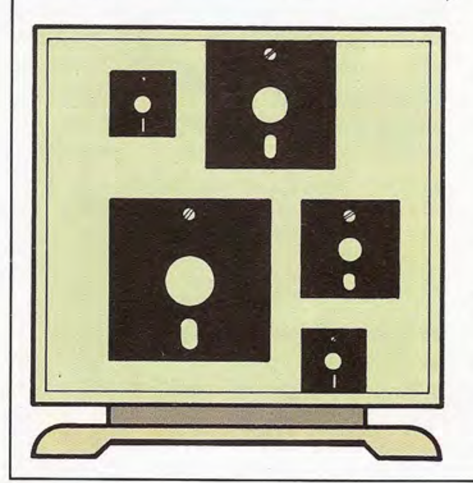
Básicamente, se orientan en dos direcciones: la que contempla la gestión de los diferentes discos asociados al sistema, y la relativa a la gestión de la impresora.

Dentro del primer grupo se encuentra el comando BACKUP, cuya misión es la de copiar el contenido completo de un disco a otro o bien a una cinta magnética. Tal operación conviene realizarla a intervalos de tiempo regulares, en orden a garantizar la seguridad de la informa-



La distribución de ficheros en disquete

Cuando un usuario crea, copia o borra un fichero en un disquete, se desencadena un proceso variable dependiente de la orden que ejecute. Sin embargo, existe una estrecha relación entre las tres operaciones; una relación fundamentada en el uso que hacen del espacio disponible en el disco.



Cuando se parte de un disquete virgen y se van creando ficheros sobre el mismo, estos ficheros se irán ordenando uno tras otro sin dejar entre ellos espacios vacíos. Ello significa que el tamaño máximo de cualquier fichero que se desee introducir posteriormente, coincidirá con la diferencia entre la capacidad total de almacenamiento del disquete y el espacio ya ocupado por otros ficheros. Cuando se borra un fichero, el sistema operativo toma el área de memoria ocupada por éste y la pone a disposición de otros datos, aunque no procede a ninguna reordenación de los ficheros sobre el disco. Ello se traduce en que el tamaño máximo de cualquier fichero que desee introducirse quedará limitado al espacio de la mayor área libre que se encuentre entre los ficheros residentes, y no por la diferencia entre «área utilizada» y «área total de memoria».
De ello se desprende que un disquete puede contener muy pocos ficheros y de pequeño tamaño, con lo que teóricamente el espacio de memoria disponible será grande y, sin embargo, por estar estos ficheros espaciados por el disquete, el tamaño máximo del fichero almacenable puede resultar muy inferior.



La vía de diálogo de cada usuario la aporta el terminal: teclado más pantalla. A través de este periférico de entrada/salida, el operador explota las aplicaciones y accede a los recursos del sistema bajo el control del OASIS.

ción obteniendo periódicamente copias de seguridad. Su formato es el siguiente:

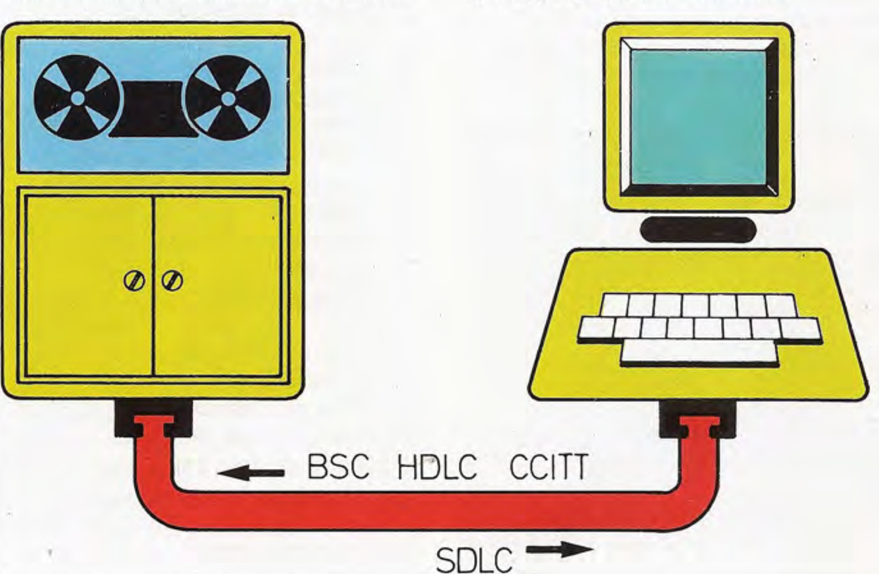
BACKUP (d1 d2) (NOVERIFY)

d1 y d2 son, respectivamente, las etiquetas de los directorios (discos o volúmenes) correspondientes al origen y destino de la información a copiar. El parámetro NOVERIFY, en el caso de ser incluido, señala que la copia no debe ser comprobada.

SPOOLER es un comando destinado al control del trabajo con impresora; este puede ser utilizado siempre y cuando el hardware lo soporte. Por medio del mismo es posible gestionar las colas de impresión que se generan en el sistema al enviar los diferentes usuarios sus listados hacia la impresora.

El acceso a los grandes ordenadores

Cada vez es más frecuente el uso de los microordenadores como nexo de unión con grandes ordenadores o «main frames», de modo que los usuarios modestos tengan acceso a la gran velocidad de proceso de que disfrutan los grandes ordenadores. En tal situación, los microordenadores se destinan principalmente a la preparación de datos y a la recogida de resultados, ya que el tratamiento es bastante más fácil en estos últimos debido a la gran cantidad de software existente en el mercado orientado hacia este objetivo (hojas electrónicas, paquetes de gestión, gráficos, etc.). Para la comunicación con el gran ordenador es necesario disponer de una conexión física (un dispositivo de interface serie de tipo RS232-C o RS422Z y cable coaxial es suficiente en la mayoría de los casos), además del software adecuado para emular los diversos protocolos utilizados en la transmisión de información. La forma en que se transfiere la información de un sistema a otro es puramente secuencial, un bit a continuación de otro, y tiene dos variantes básicas: síncrona y asíncrona. El nombre de transmisión síncrona procede de que la información es enviada en paquetes discretos, con intervalos de tiempo constante entre paquetes. Una velocidad de transmisión típica es, por ejemplo, la de 9.600 baudios.



Por contra, la transmisión asíncrona se realiza en intervalos de tiempo no constantes. Cada paquete de bits es identificado por ciertos bits especiales, de control, que señalan el comienzo y el fin de la transmisión. Este segundo método presenta la ventaja de que no requiere software ni equipos tan sofisticados como los necesarios para la transmisión síncrona; aunque tiene el inconveniente de que la comunicación es más lenta debido a que un extremo no sabe cuándo va a recibir información del otro.

La información puede ir de un extremo a otro de la línea en un solo sentido (half-duplex) o en ambos (full-duplex).

Existen métodos de control para la detección de errores, como por ejemplo, el derivado de utilizar 7 bits para la transmisión y un octavo para reflejar su paridad (si la paridad se define como par y el número de bits cuyo valor es uno es impar, este último bit será 1, y 0 en caso contrario).

La norma más extendida para la transmisión de datos es el SNA (System Network Architecture) de IBM, con sus protocolos BSC (Bysinchronous System Communication), SDLC (Sinchronous Data Link Control) y HDLC (High-level Data Link Control), así como las normas CCITT. La puesta a punto de los diversos sistemas y equipos para llevar a cabo la transmisión de datos a través de una línea (línea de transmisión, controladores, emuladores de protocolo, etc.) puede conducir a gastos de hasta 6 y 7 veces el valor del microordenador con el que se pretendía realizar la conexión. De ahí que también se esté impulsando la vía de la conversión de terminales clásicos de gran ordenador en terminales inteligentes, con las mismas propiedades y periféricos que los microordenadores, evitando así la parte más cara constituida por las comunicaciones.

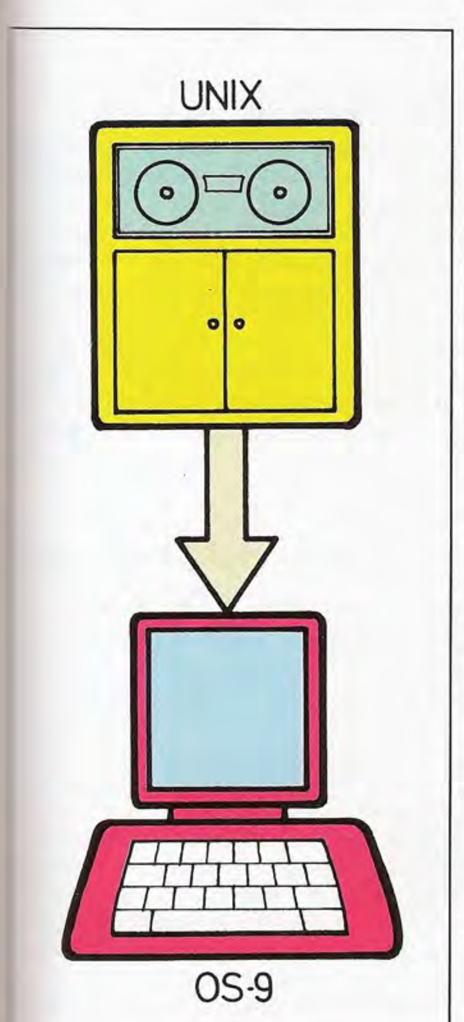
OS-9

Un potente sistema operativo para pequeños equipos



A través del estudio de los diferentes sistemas operativos abordados en esta obra, ha habido ocasión de comprobar las diversas formas de atacar el problema de la gestión de los recursos informáticos del ordenador; tanto por lo que se refiere a periféricos, como a programas de aplicación, así como a la interface ordenador-usuario. Cada uno de los sistemas operativos tratados resolvía el problema con ciertas peculiaridades. Uno de ellos llegó a ser señalado como el sistema operativo tal vez con más posibilidades de futuro entre los destinados al ámbito de los microordenadores de tipo profesional. Se trataba del sistema operativo UNIX.

Una de las grandes ventajas del UNIX residía en su transportabilidad y en su alta capacidad para adaptarse a otros ordenadores, debido al hecho de que está escrito en un lenguaje de alto nivel: el «C». Una muestra de esta posibilidad la aporta el sistema operativo OS-9: un descendiente directo del UNIX aunque adaptado a microordenadores destinados al mercado doméstico y educativo, como es el caso del popular DRAGON.



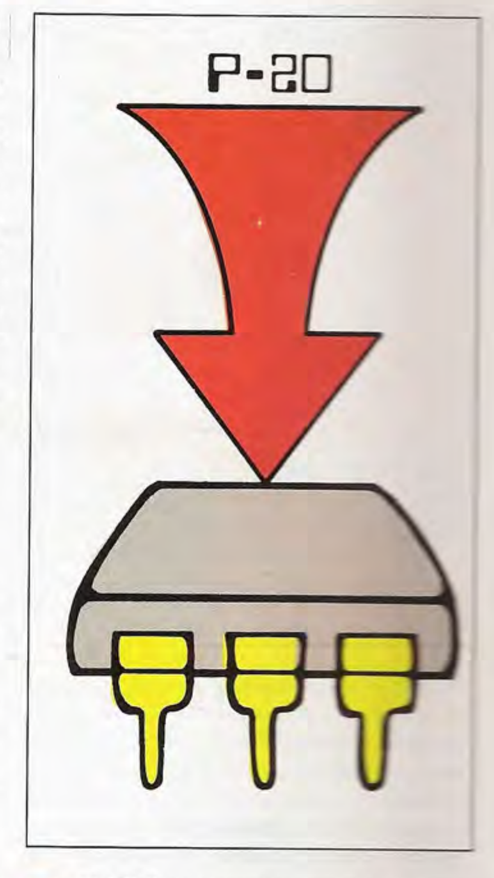
El sistema operativo OS-9 es un descendiente directo del popular y potente UNIX.

Introducción al OS-9

El sistema OS-9 es un sistema operativo multitarea y multiusuario: es decir, permite la existencia de varios usuarios trabajando simultáneamente con el ordenador, o bien la ejecución de varios programas funcionando concurrentemente. Como soporte del mismo se emplea un equipo basado en un microprocesador Motorola 6809; precisamente, el empleado en el DRAGON.

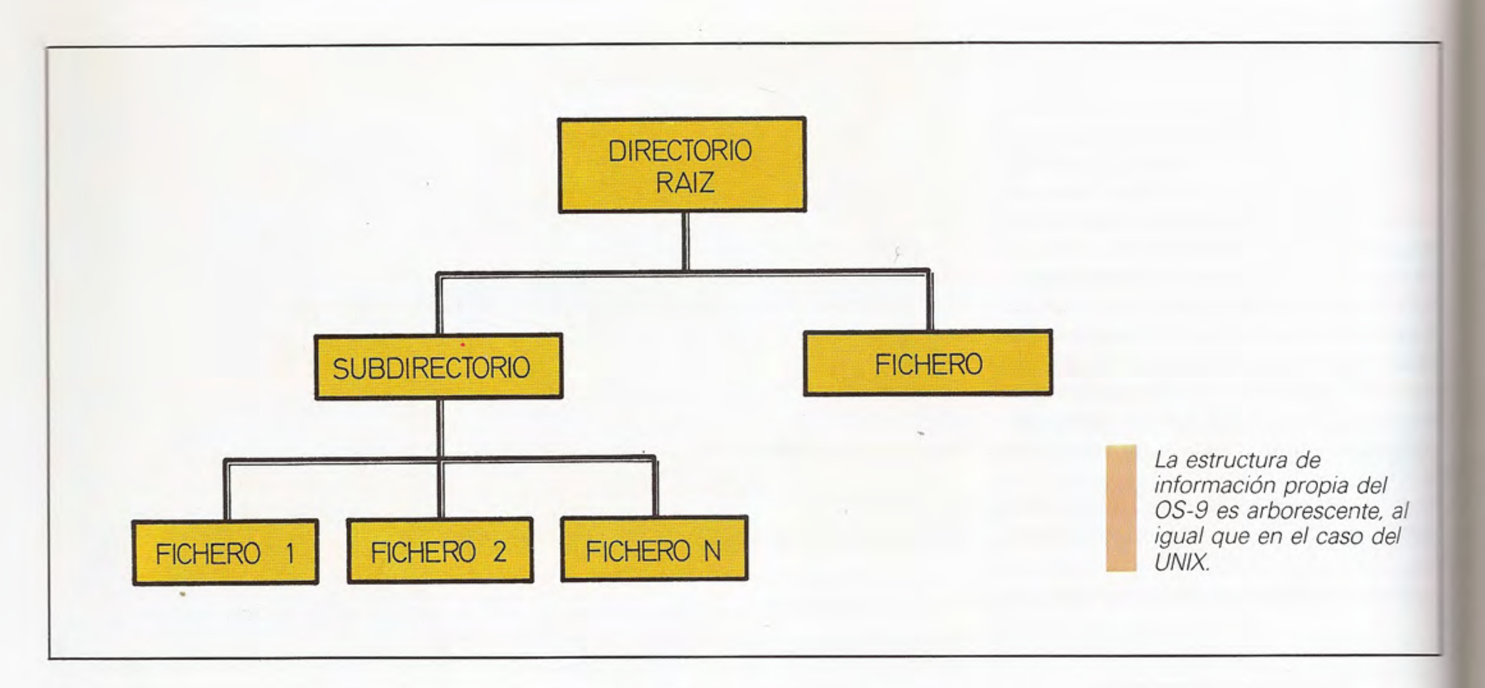
El origen de los requerimientos de este sistema operativo se encuentran en el desarrollo por parte de Microware de un nuevo dialecto del lenguaje BASIC con la colaboración de Motorola durante los últimos años de la década de los setenta. El desarrollo de este lenguaje presuponía la existencia de un potente sistema operativo que soportara las peticiones que le pudieran ser impuestas por el lenguaje. El OS-9 se inspiró plenamente en el sistema operativo UNIX de Bell Laboratories. La copia no fue directa, toda vez que las necesidades de un pequeño microordenadores, así como sus disponibilidades hardware, no son las mismas que las propias de los ordenadores sobre los cuales el UNIX se desarrolló inicialmente. Las diferencias fundamentales del OS-9 con respecto al UNIX son las que se detallan a continuación:

- El OS-9 ha sido diseñado para un entorno de memoria en el que conviven zonas de RAM y ROM, soportando código reentrante de forma más efectiva.
- El lenguaje en el que está escrito el OS-9 es el ensamblador correspondiente al microprocesador 6809 en lugar de lenguaje de alto nivel C del UNIX, con lo cual se consigue mejorar el rendimiento del producto resultante.
- Durante la operación en régimen multitarea/multiusuario, el OS-9 no gestiona el almacenamiento dinámico en disco de los programas en ejecución, sino que los conserva en memoria en todo momento.



El OS-9 gestiona las tareas de los distintos usuarios fundamentalmente a través de la memoria interna del equipo.

Todas estas diferencias son consecuencia lógica de la diferencia de entornos en los que se explotan estos sistemas operativos. El UNIX fue diseñado para ordenadores como por ejemplo PDP-11 de Digital, equipos con una potente CPU, abundancia de memoria central y de sistemas de almacenamiento secundario de gran capacidad y reducidos tiempos de acceso. En estas condiciones resulta lógico que el OS-9 mantenga continuamente en memoria los programas, toda vez que los tiempos de



La estructura de información propia del OS-9 es arborescente, al igual que en el caso del UNIX.

acceso a los disquetes que constituyan el medio de almacenamiento principal de los microordenadores y sobre los que está grabado el OS-9, son muy largos en comparacion con el tiempo medio de acceso a los discos de los ordenadores sobre los cuales reside normalmente el UNIX. Esta diferencia hace ilógico cualquier intento de paginación, dado que produciría en los tiempos de respuesta del equipo un retraso inaceptable para el usuario.

Por razones similares, cabe señalar como lógica la utilización de un lenguaje ensamblador para el montaje del sistema operativo, con preferencia a un lenguaje de alto nivel. Con la primera opción se consigue mejorar la respuesta a las peticiones del usuario, al disminuir los requisitos de ciclos de máquina exigidos a la CPU.

GESTOR DE ENTRADA / SALIDA
GESTOR DE FICHEROS EN DISCO
GESTOR DE FICHEROS DE CARACTERES
ACCIONADOR DE DISCO
ACCIONADOR DE DISCO
ACCIONADOR DE DISCO
ACCIONADOR DE IMPRESORA
ACCIONADOR DE TERMINAL
DESCRIPTOR DE DISPOSITIVO
DESCRIPTOR DE DISPOSITIVO
DESCRIPTOR DE DISPOSITIVO
DESCRIPTOR DE DISPOSITIVO

Estructura del sistema de gestión de E/S del OS-9.

Necesidades hardware

Como requisitos mínimos para dotar a un equipo del sistema operativo OS-9 se necesitan 4 Kbytes de memoria ROM y 2 Kbytes de memoria RAM. Hay que tener en cuenta que el OS-9 está compuesto por una serie de módulos que se organizan y enlazan automáticamente en el momento de arrancar el sistema. Esta estructura permite la reconfiguración del sistema sin más que cargar en memoria los módulos necesarios.

A efectos de desarrollo, la configuración típica mínima es la señalada a continuación:

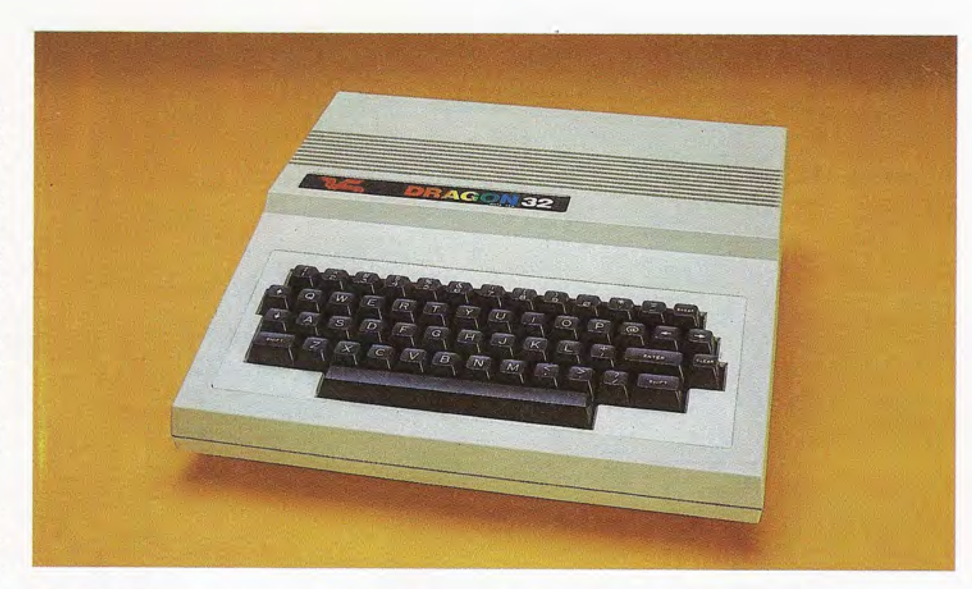
- 64 Kbytes de memoria RAM, dividida entre la necesaria para la programación en lenguaje de alto nivel y la necesaria para la programación en lenguaje ensamblador.
- 4 Kbytes de memoria en ROM, de los cuales 2 Kbytes deben estar direccionados en $F800-$FFFF, mientras que los restantes son independientes en su posición.

A efectos operativos de ampliación de la capacidad de memoria del ordenador, el controlador de entrada/salida del ordenador conviene que esté localizado en una posición de memoria los más alta posible; de esta forma será posible incrementar la capacidad de RAM del equipo con menores problemas para el usuario.

La estructura interna del OS-9

La estructura de este sistema operativo consta de un conjunto de módulos, cada uno de los cuales está destinado a cumplir una serie de funciones específicas. Los módulos son seleccionados en función de las necesidades del equipo sobre el que va a ser implementado, de manera que si un ordenador no tiene unidades de disco no necesitarán los módulos de OS-9 relacionados con el control de dichas unidades.

La base del sistema son los módulos NUCLEO, RELOJ e INIC, El primero, como su propio nombre indica, proporciona los servicios principales de control del sistema, como son la gestión de la memoria, de los programas en curso y el enlace con los restantes módulos

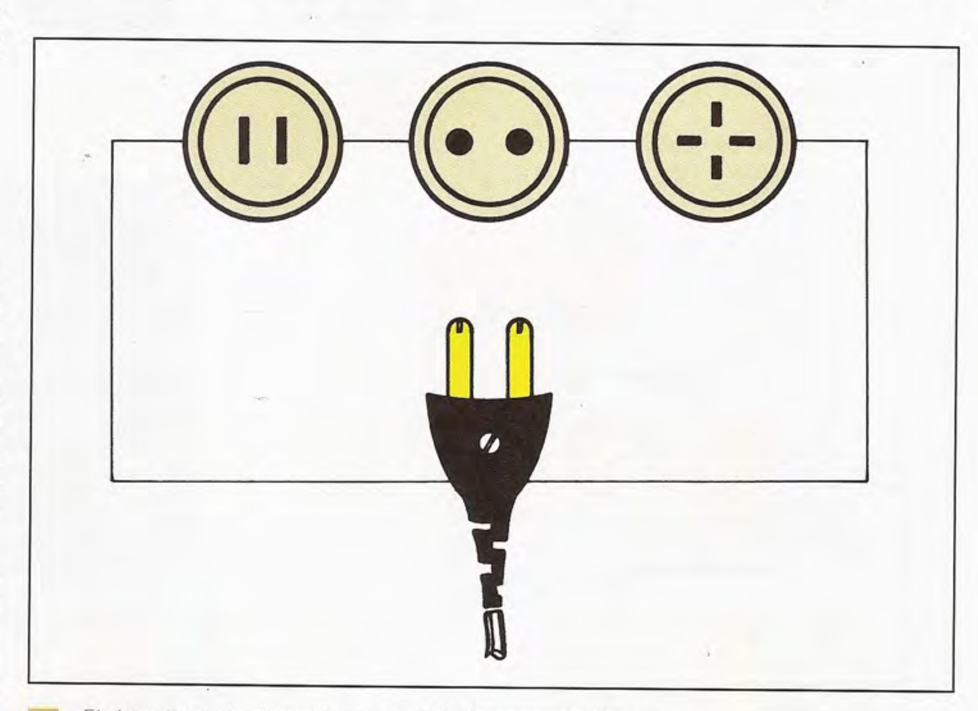


El OS-9 es un sistema operativo concebido para pequeños microordenadores basados en el microprocesador 6809 de Motorola.

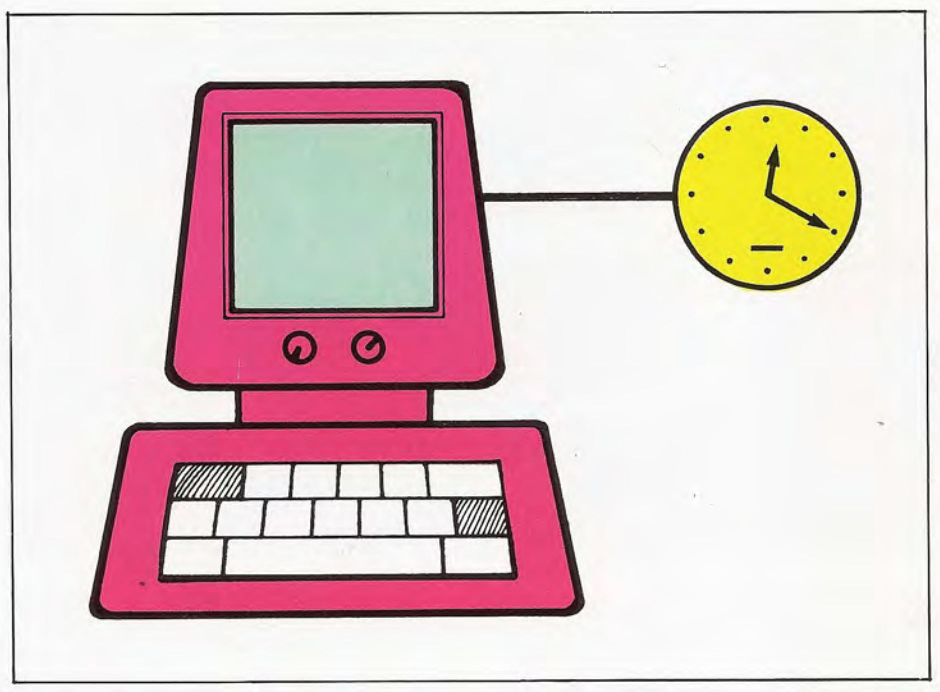
del sistema. El módulo RELOJ gestiona el reloj interno del sistema para la adecuada operación del software en tiempo real. El módulo INIC es una tabla que contiene los parámetros de inicialización del sistema, y es utilizada por el núcleo durante el proceso de arranque, asignándose a través de la misma valores tales como los nombres de los dispositivos del sistema.

En un nivel inmediatamente inferior al módulo INIC, se encuentra el gestor de entrada/salida.

En el siguiente nivel aparece el gestor de ficheros, encargado de gobernar las peticiones de entrada/salida a tra-



El descriptor de dispositivos controla la conexión física de los periféricos al microordenador.



El módulo reloj controla los procesos en tiempo real ejecutados por el sistema.

vés de periféricos. Este se divide en dos zonas. Por un lado se encuentra el denominado «gestor de ficheros para bloques aleatorios» el cual procesa todas las operaciones relacionadas con unidades de disco, y por otro lado aparece el gestor de ficheros de caracteres secuenciales, el cual trabaja sobre dispositivos de entrada/salida de naturaleza secuencial, tales como impresoras y terminales.

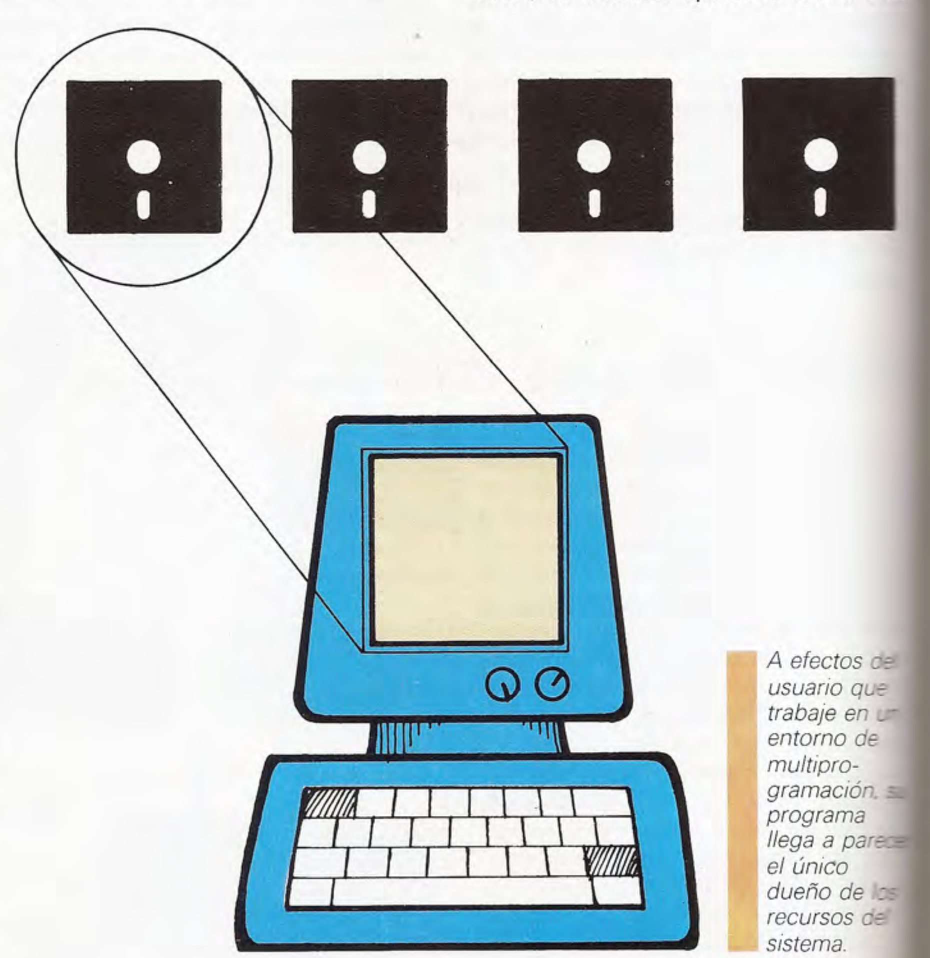
El hecho de que se hable de gestores de ficheros se debe a que, al igual que ocurre con el UNIX, el OS-9 trata a los dispositivos de entrada/salida como si de ficheros se tratase.

En un nivel inferior y siguiendo una jerarquía descendente por lo que se refiere al control de periféricos se encuentran las denominadas unidades de accionamiento de periféricos. Su cometido reside en el control de actividades físicas básicas de entrada/salida con el hardware específico. A través de estas unidades de accionamiento es posible preparar gestores específicos para controlar elementos hardware distintos de los instalados inicialmente en el ordenador. El nivel final en la gestión de entrada/salida corresponde al descriptor de dispositivo. Estos descriptores están relacionados directamente con la conexión física y son pequeños ficheros en los que se almacenan los datos correspondientes a las puertas de entrada/salida. Almacenan, por ejemplo, los nombres lógicos y la dirección física de las puertas, así como otros datos de inicialización necesarios.

Se observa, pues, que la estructura del sistema de comunicaciones de entrada/salida es jerárquica en su alcance, reduciéndose desde una óptica amplia a otra más específica. Este tipo de estructura hace que resulte muy fácil controlar una amplia gama de periféricos. En el momento que el usuario lo considere oportuno pueden añadirse al sistema los módulos necesarios para aumentar el número o variar las características de los periféricos ya instalados.

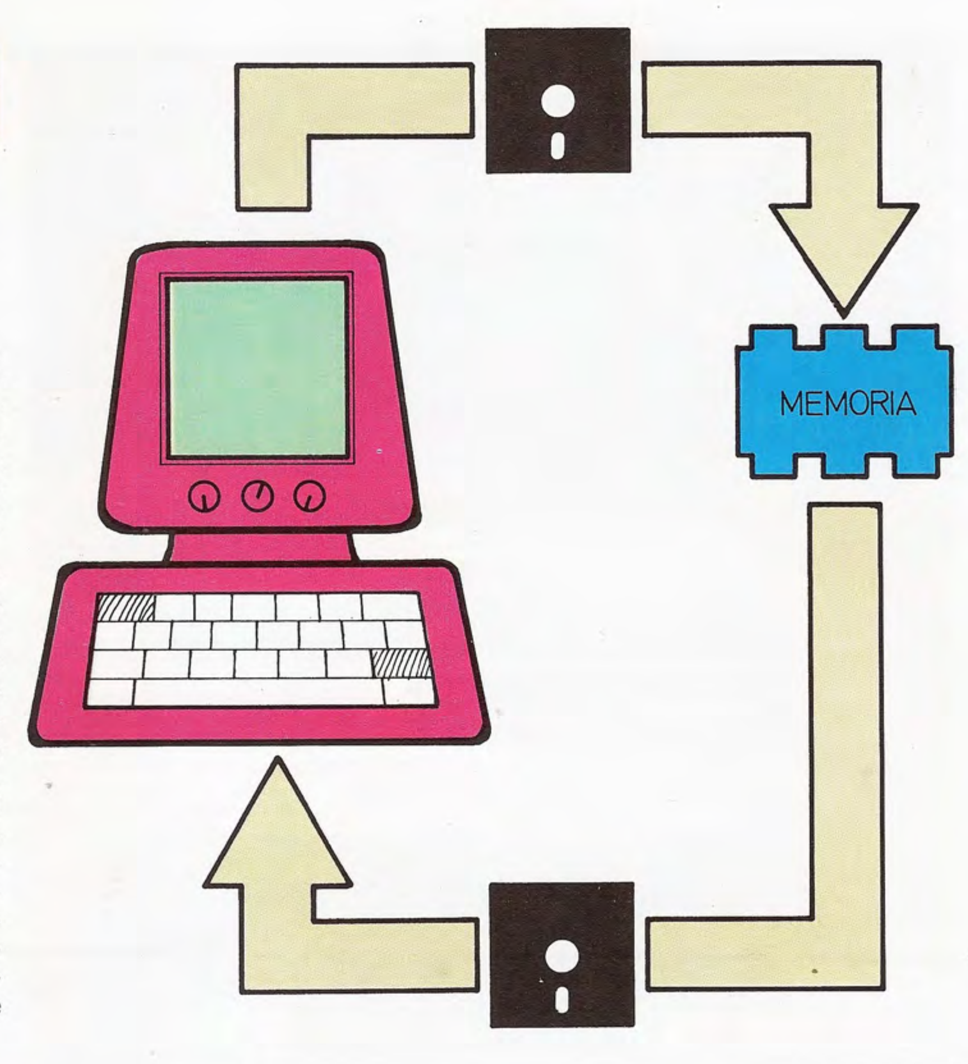
Control de la CPU en multiprogramación

Debido a su directa descendencia del UNIX, el sistema operativo OS-9 está



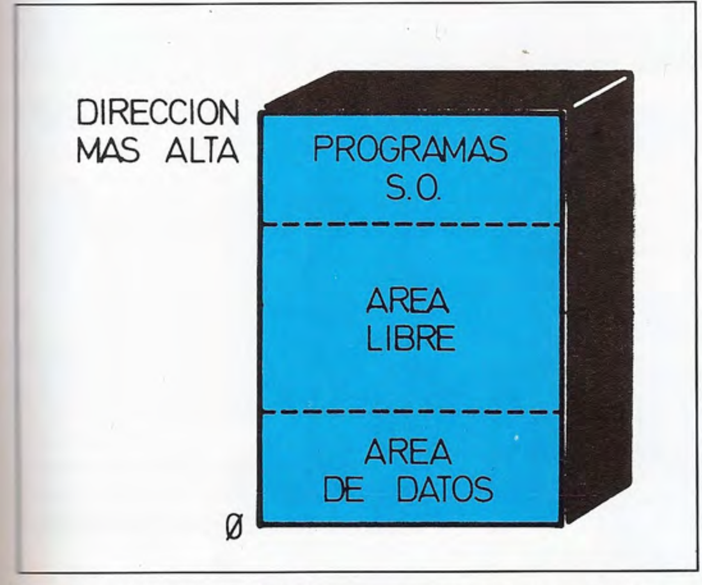
A efectos del usuario que trabaje en un entorno de multiprogramación, su programa llega a parecer el único dueño de los recursos del sistema.

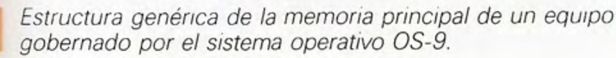
En el ámbito del OS-9, los programas en operación multiárea se mantienen continuamente en la memoria central del ordenador.

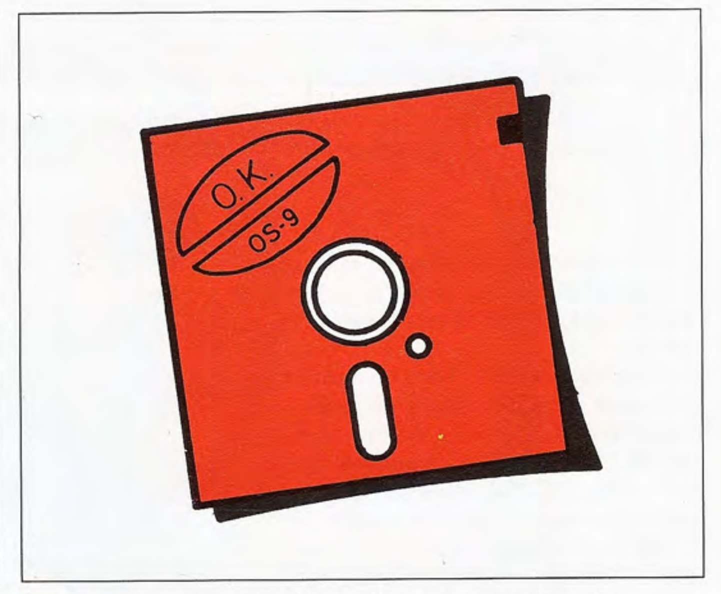


perfectamente capacitado para trabajar en el régimen denominado de multiprogramación. En esta modalidad de operación, también conocida como tiempo compartido, el OS-9 puede mantener más de un programa funcionando en el ordenador.

El procedimiento seguido normalmente por los ordenadores para operar en esta modalidad, es asignar un tiempo de CPU determinado a cada uno de los programas que están ejecutándose en su interior. Cuando llega el momento de que un determinado programa pase a ocupar los recursos de la CPU, éste es cargado en la memoria principal desde un área de almacenamiento especial, localizada en uno de los periféricos de almacenamiento masivo asociados al ordenador, mientras el programa anteriormente en curso, es «descargado» de la memoria central del ordenador y depositado en la referida área de almacenamiento masivo.




Estructura genérica de la memoria principal de un equipo gobernado por el sistema operativo OS-9.



El sistema operativo OS-9 sólo asigna recursos a los programas en disposición de utilizarlos.



Los microordenadores de la gama DRAGON cuentan con una versión del sistema operativo OS-9.

Por medio de esta técnica, es posible mantener varios programas ejecutándose simultáneamente sobre el mismo equipo.

En el caso del OS-9 nos encontramos con una situación diferente, toda vez que el entorno para el cual ha sido creado, el de los microordenadores, posee unidades de almacenamiento masivo relativamente lentas y de escasa capacidad para permitir su empleo en esta modalidad de funcionamiento. Es por ello que el OS-9 se ve obligado a mantener en la memoria principal del ordenador a todos los programas en curso, y gestiona internamente su asignación a los recursos de la CPU.

Para cumplir con estas funciones, el sistema operativo tiene que controlar los siguientes factores:

— Tiempo de CPU
— Mapa de memoria y recursos disponibles de la misma
— Sistema de Entrada/Salida

Los grandes problemas derivados de un inadecuado control de tiempo de CPU, tienen su origen en el hecho de olvidar una regla fundamental del trabajo con ordenadores: que la operación de la CPU es muchísimo más rápida que la del ser humano o de los dispositivos de entrada/salida a través de los que comunica. Así, es frecuente que se pierda un tiempo precioso esperando que un programa gestione una operación de entrada/salida de información, en la que la CPU juega un papel muy restringido, puesto que la mayor parte del trabajo lo realiza el controlador del periférico con el que comunica.

Como ejemplo típico de lo que acabamos de señalar cabe citar a los programas de tipo interactivo en los cuales mientras que el usuario no introduzca la información necesaria para activar el proceso no puede llevarse a cabo ninguna tarea, con el consiguiente desperdicio de recursos. El sistema operativo OS-9 hace uso de una técnica denominada «time slicing», por la cual divide el tiempo de CPU tan sólo entre aquellos programas que están en condiciones de hacer uso de sus recursos. Este proceso se lleva a cabo mediante el empleo de útiles especiales de hardware y software, cadenciados a partir del reloj de tiempo real del equipo. Las divisones que se hacen del tiempo de CPU dependen de la prioridad relativa de los programas en ejecución, ya que, lógicamente, a mayor prioridad tanto mayor será el tiempo de CPU que se asigne al proceso en cuestión.

En términos generales, el tanto por ciento de recursos otorgados a un programa viene definido por la expresión:

Cuota de CPU = Prioridad del programa/suma de prioridades

Este procedimiento permite que los programas funcionen de manera casi simultánea, dando la impresión de que cada programa es el único poseedor de los recursos del sistema. Lógicamente, hay una penalización de tiempo puesto que a fin de cuentas son varios los programas que se reparten la atención de la CPU.

Procesos en multiprogramación

A efectos de operación, el tiempo de CPU se asigna a los programas en fun-

ción que se encuentren en un estado activo; esto es: en disposición de ejecutar funciones. Los diferentes estados de proceso son.

• Activo: este estado es el correspondiente a aquellos procesos que están llevando a cabo operaciones efectivas, siendo los únicos a los que se les asignan recursos de CPU.

• De espera: estado que corresponde a aquellos procesos cuya ejecución se encuentra suspendida, pendiente de que finalice otro programa.

• Durmiente: corresponde a aquellos procesos que han sido suspendidos a requerimiento propio, en espera de alguna señal predeterminada que los haga volver al estado activo.

Para poner en funcionamiento un nuevo proceso y asignar inicialmente los recursos que ha de consumir, es necesario efectuar una serie de operaciones. En el caso del sistema operativo OS-9 éstas se realizarán de manera automática apelando a la función FORK. Esta controla el proceso, y si no puede darle curso, transfiere un código de error al proceso a partir del cual se originó, de manera que éste pueda decidir las acciones posteriores a llevar a cabo. Cuando un determinado proceso tiene que generar nuevos procesos encadenados al mismo, ha de comunicar al sistema operativo las siguientes informaciones.

— Módulo primario: es también el nombre con el cual se conocerá al programa que ejecutará el nuevo proceso. Este puede cargarse en memoria desde un medio de almacenamiento externo, o bien, puede encontrarse ya residiendo en memoria.

• Parámetros: son los datos que necesita el nuevo proceso para su correcta operacion.

• Número de usuario: en él se fundamenta la técnica empleada por el sitema de seguridad para distinguir y diferenciar los procesos pertenecientes a un usuario específico.

— Prioridad del proceso: permitirá determinar la proporción de tiempo de CPU que se otorgará al proceso lanzado. Normalmente, la prioridad será la misma que la del proceso desde el que se lanzó.

Gestión de memoria

Cuando se arranca el sistema operativo, la primera operación que se lleva a efecto es la asignación por parte del sistema de los recursos de memoria. En la parte más alta de la misma se cargan los distintos módulos del sistema operativo, la zona inferior de la memoria se destina a datos y la zona intermedia queda vacía para su posterior uso en la gestión de nuevos programas y procesos.

La memoria central del equipo es gestionada por el OS-9 de manera que se opere con los programas de forma reentrante, es decir, un mismo módulo de programa puede ser empleado por varios usuarios, si bien, y dado que cada

El ordenador rentable

Una afirmación plenamente extendida en nuestros días es que el ordenador es una máquina maravillosa capaz de efectuar casi cualquier tarea que le sea encomendada. Mucho de verdad hay en ello ya que, efectivamente, si un ordenador se programa adecuadamente puede llevar a cabo multitud de tareas con *mayor o menor* efectividad. Y ahí se encuentra la clave. Un ordenador puede gestionar igualmente la nómina de una empresa con diez mil empleados, como la de otra empresa con tan sólo tres empleados. No obstante, en este último caso el coste del equipo así como el de las operaciones necesarias para que se ejecute la referida actividad, pueden hacer mucho más rentable para el usuario adoptar un proceso manual.
El ordenador es también altamente rentable en el control de procesos industriales: puede controlar en tiempo real la evolución de un gran número de señales, y tomar decisiones para la mejor operación y rendimiento del conjunto global. Sin embargo, el ordenador no será rentable si su actividad se reduce a controlar el encendido y apagado de las luces exteriores de una casa.
En términos generales, las tareas que un ordenador puede llevar a cabo con mayor rendimiento son las que comparten alguna de las siguientes características:

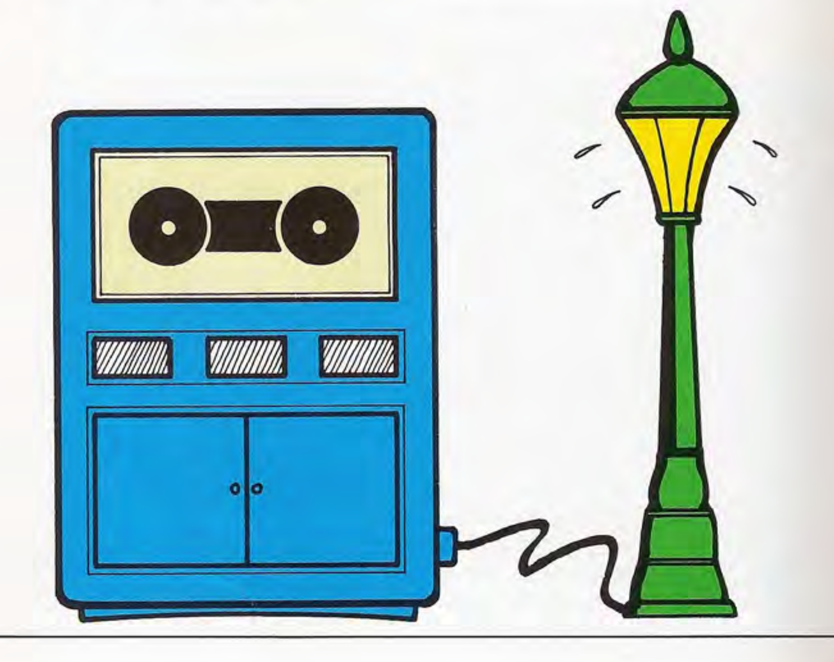
• Hay que manipular y procesar grandes volúmenes de datos residentes en ficheros.

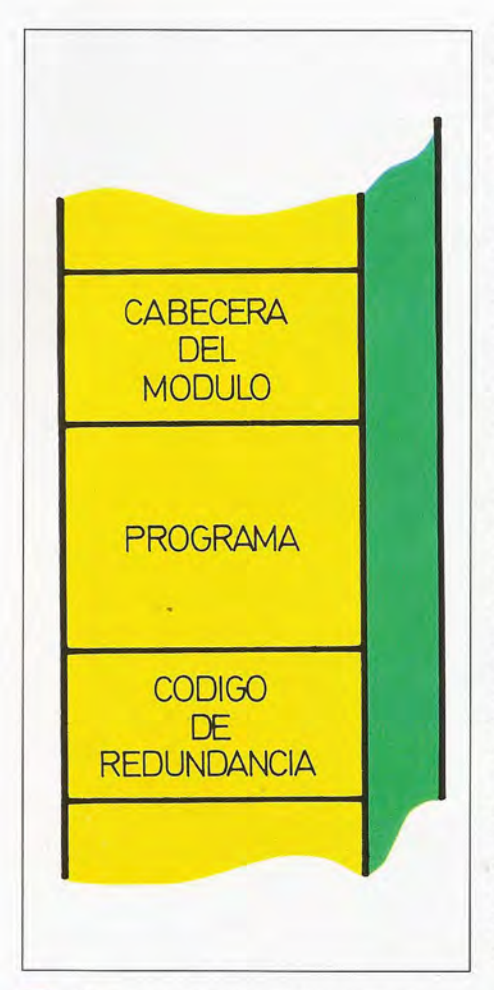
• Deben controlarse procesos repetitivos que pueden tomar ventaja de la velocidad de proceso del ordenador

• Procesos de gestión en los que hay que realizar un gran número de transacciones.

• Se trata de ejecutar procesos de gestión altamente definidos.

Dada su naturaleza —una mera máquina— el ordenador no puede mostrar ninguna de las restricciones emocionales propias del ser humano: ira, fatiga, frustración... y tampoco puede hacer juicios de valor, a menos que sea programado específicamente para ello y los datos le sean suministrados de forma adecuada. De ahí que corresponda al usuario considerar los factores que determinan si el empleo del ordenador para determinada actividad se ajusta a criterios de rentabilidad, eficacia, precisión y comodidad.





Disposición de los módulos en la memoria principal.

uno de ellos puede utilizarlo de distinta forma, se destinan áreas de memoria independientes para las respectivas variables durante el tiempo de ejecución. Un ejemplo de ello lo aporta el intérprete del lenguaje BASIC-09; éste precisa 22 Kbytes de memoria y permite que varios usuarios accedan a él sin necesidad de nuevas copias, lo cual agotaría rápidamente los recursos de memoria del sistema. La gestión de los diferentes procesos que utiliza cada módulo de programa corre a cargo del sistema operativo, el cual eliminará la zona de memoria asignada una vez que los procesos hayan terminado. La asignación de memoria a los módulos de programa se realiza mediante la cabecera de dichos módulos, en la cual se indica cuál es la cantidad de memoria, tanto estática como dinámica, que ha de asignarse al proceso. La carga de programas en memoria puede hacerse por medio de la instrucción LOAD la cual, al igual que en el caso de la instrucción FORK, carga los programas únicamente si éstos no se encuentran ya en memoria. En este último caso, cargará los datos de programa ya cargados. Si el programa se encuentra ya en memoria se seguirá un proceso análogo; si bien, se incrementará un contador, denominado de encadenamiento, que en cada momento controla los procesos que están haciendo uso del módulo en cuestión.

Un factor problemático lo constituye el hecho de que los programas reubicables pueden, en efecto, ser cargados en cualquier posición de memoria, pero no pueden ser reubicados dinámicamente después. En definitiva, cuando un programa se carga lo hace en el primer espacio libre adecuado a su tamaño. En el caso de que haya varios programas funcionando puede ocurrir que alguno de ellos finalice sus operaciones y libere memoria; no obstante, los diferentes espacios así liberados, aun siendo grande su capacidad conjunta, pueden estar desperdigados de manera que sólo liberen pequeños huecos insuficientes para cargar otros programas incluso de moderada longitud. Este fenómeno se conoce como fragmentación de memoria y, normalmente, sólo se detecta a través del comando MFREE el cual señala la posición de cada área de memoria sin utilizar, así como el tamaño de la misma. Para obviar tal inconveniente, es preciso volver a cargar los módulos dispersos para la memoria, asegurándose previamente de que no están en uso o no van a ser utilizados con inmediatez.

El núcleo del sistema operativo OS-9

El núcleo del sistema operativo OS-9 actúa como gestor y supervisor de los recursos informáticos globales puestos a disposición del usuario. Reside en memoria ROM ocupando, junto con el módulo de inicialización de disco, cerca de 4 Kbytes.

Las funciones más destacables del núcleo son:

- Inicialización del sistema tras el arranque, por medio del módulo INIC
- Gestión de multiproceso e interrupciones
- Gestión de la memoria
- Control de peticiones de servicio

La inicialización del sistema se realiza cargando en memoria volátil los datos contenidos en ROM. Una vez evaluada la cantidad de memoria disponible, se cargará en la memoria principal el módulo necesario.

Tras estos pasos, se inicializan los tamaños de las tablas y los nombres de los dispositivos del sistema para, de inmediato, dar comienzo las operaciones de explotación.



Mapa de memoria típico de un microordenador equipado con el sistema operativo OS-9.

Pick

Un sistema operativo amigo del usuario



Recientemente un consultor de ordenadores atrajo la atención en el comercio revelando uno de sus secretos para su éxito en el trabajo. Parece ser que este profesional del procesamiento de datos comenzaba por aconsejar a sus clientes que se deshicieran de sus ordenadores. Muchas de sus compañías pasaron sin ellos y sus cuotas y reputación continuaron creciendo.

Esto es penoso, pero no del todo sorprendente. Desde principios de los 70 existe un concienciamiento por parte de los profesionales de procesamiento de datos de que los ordenadores dificultan la buena marcha de los negocios.

Todo comienza cuando un hombre de negocios tiene que hacerse especialista en el procesamiento de datos, o buscar a alguien, para sacar el máximo provecho a su ordenador. Los sistemas eran solamente productivos cuando se conocían las técnicas de procesamiento de datos. Durante los años setenta comenzó a estar claro que: los sistemas debían ser útiles para el equipo de gestión, debían ser menos especializados, más orientados hacia los negocios y, en general, más asequibles. Esta es la única forma de que los profesionales del procesamiento de datos puedan encontrar facilidades en vez de obstáculos.

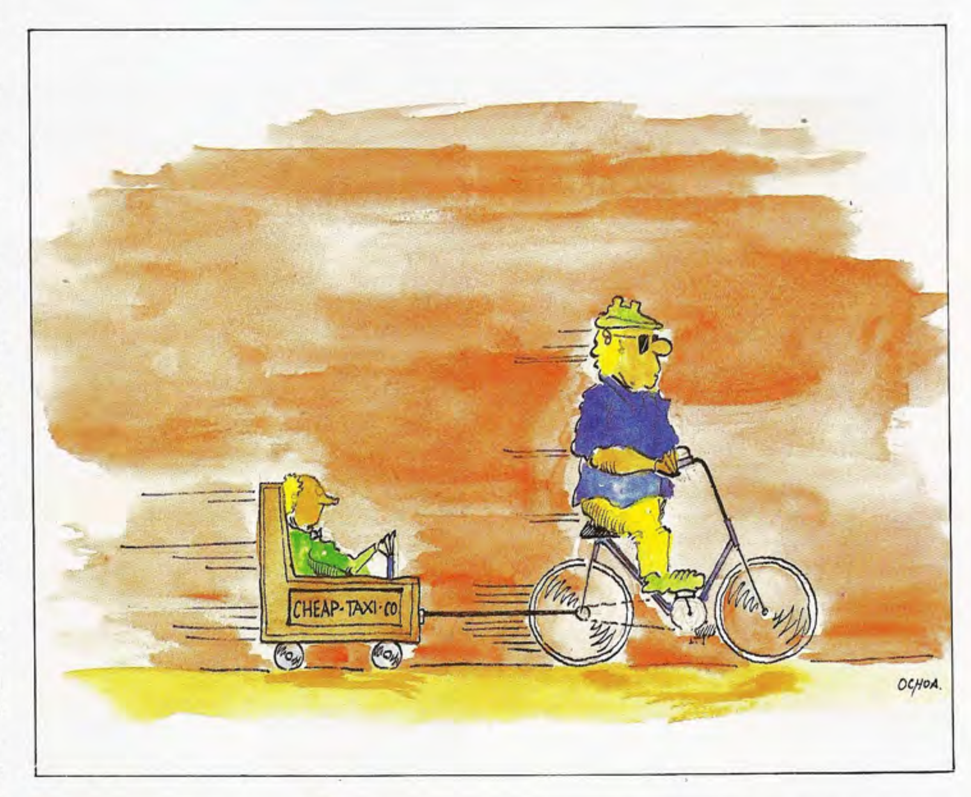
La situación está mejorando y los ordenadores, por sí mismos, son parte de esta mejora. La industria de los ordenadores ha enfocado sus esfuerzos hacia los sistemas «amigos de los usuarios». Un sistema «amigo del usuario» es aquel que relaciona la forma de trabajo del usuario con los datos. Esto incluye procesos no técnicos fáciles de aprender y de manejar. Con ellos el hombre de negocios puede concentrarse más en los problemas de la empresa y menos en el ordenador.

Esta idea fue la base para un acercamiento a finales de los 60, lo cual dio lugar a un sistema que permitía todo el manejo de datos automáticamente, permitiendo al usuario almacenar y recuperar información refiriéndose a ella con un nombre. El avance se incrementó con el trabajo interactivo ordenador-usuario, usando simples instrucciones inglesas en una pantalla CRT. El sistema se fue desarrollando hasta llegar a ser independiente del hardware. Fue diseñado para correr eficientemente en ordenadores pequeños que compartían recursos de ordenador, tales como impresoras y memoria. El sistema operativo debería asignar el espacio de memoria, manejar ficheros, organizar y reorganizar dinámicamente el espacio en disco, de tal forma que el usuario pudiera concentrarse en sus datos en vez de en el sistema de procesamiento.

Don Nelson y Dick Pick usaron todo ésto para crear el sistema operativo que hoy se conoce con el nombre de Pick. Este se comercializó por Microdata Corporation en 1974, y se ha usado con éxito desde entonces en docenas de ordenadores grandes y pequeños. En 1984 el sistema operativo Pick se utilizó en ordenadores IBM XT e IBM 4300, com-



En los años sesenta adquirir un ordenador para gestión podía suponer un problema más, en vez de lo que se esperaba de ellos.



Las máquinas estaban tan orientadas al hardware que para usarlas era necesario ser experto en procesamiento de datos o contratar a alguien que lo fuera.



La tecnología actual tiende a hacer los sistemas más accesibles. Lo importante deja de ser la máquina para dejar paso a los usuarios.

pletando el espectro desde micro a «mainframes».

Pick es más que un sistema operativo de una década; es un entorno de usuario avanzado, que puede cambiar la forma en que los datos son percibidos y manipulados. Además, con la memoria virtual, la base de datos relacional, el sistema orientado al usuario puede hacer época en esta generación de ordenadores.

El futuro y Pick

El sistema Pick es sin duda una visión de futuro en el entorno operativo. Ya se anticipó a su época cuando fue creado a finales de los 60, y todavía hoy es uno de los más avanzados. Su inusual longevidad es debida a una buena interfase de usuario y a la capacidad fundamental de su estructura de datos única. Además, es flexible y fácil de aprender y usar.

Pero, ¿por qué es importante Pick? Los usuarios del mañana exigirán la facilidad de aprendizaje y de uso, tanto de un micro como de un «mainframe». Millones de usuarios con sus billones de pesetas serán los que probablemente moldeen la industria del ordenador con sus demandas. Y viceversa, lo que ellos aprendan del sistema operativo Pick podrá perfilar sus propias demandas.

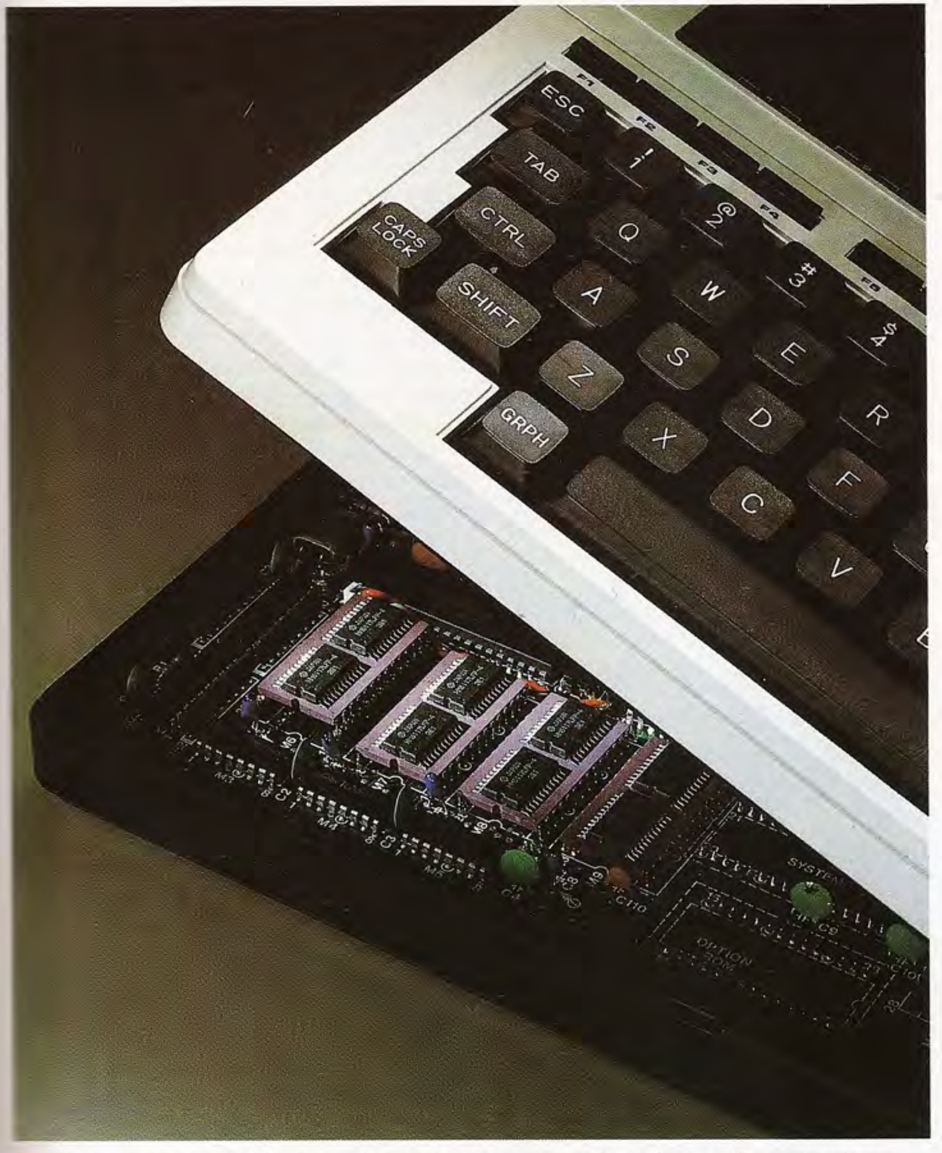
Para entender el efecto de Pick vamos a estudiar la situación actual. Durante los últimos 20 años el mercado de la informática ha sido inundado con nuevos ordenadores y nuevas aplicaciones. El sistema operativo Pick nació en los comienzos de este desbordamiento. Fue desarrollado en tiempos en que la memoria era cara. En su núcleo tenía un esquema de direccionamiento de memoria virtual que usaba la memoria disponible hasta el máximo de su eficiencia. A lo largo de los años, los avances técnicos fueron proporcionando gradualmente cantidades de memoria principal mayores; no obstante, las técnicas para el uso de esta memoria habían sido avanzadas desde tiempos en que la memoria era cara. Debido a que el esquema de memoria virtual de Pick usa la memoria mucho más eficientemente

que cualquier otro sistema, todavía hoy sigue en la brecha.

Además, hay algo más que una mera eficiencia en el trabajo. La industria tiende cada vez más a hacer ordenadores de uso más sencillo. Ya que los avances de la técnica están disminuyendo el tamaño y coste de los equipos, los ordenadores empiezan a ser más asequibles a los usuarios. Estos últimos contribuyen aún más a la modernización de los sistemas operativos con su inteligencia y dinero. La innovación es cada vez mayor y el conocimiento por parte de los usuarios también.

La demanda actual tiende a favorecer más al usuario que al técnico. El sistema Pick, con su orientación al usuario, cumple muchos de los requisitos exigidos. Este presenta una interface de usuario sencilla de aprender y un lenguaje de generación semejante al inglés hablado. Tiene un manejador de base de datos integral que soporta ficheros horizontales, jerárquicos, o de bases de datos relacionales. El sistema incluye una avanzada versión orientada a base de datos del lenguaje de programación BASIC con compilación automática y montador. Las periferias del sistema están controladas por un completo conjunto de comandos para dispositivos (construidos en la interface del usuario). Finalmente, estas características son compatibles tanto con ordenadores personales como con ordenadores medios. La portabilidad es inherente a este sistema. La demanda de tales características está creciendo en el colectivo de usuarios que comienza a tener parte activa en la informática.

¿Y cómo demandarán el crecimiento. De una forma gradual o explosiva? Nosotros pensamos que es más probable un cambio gradual de los sistemas operativos, siempre orientado hacia el usuario. La arquitectura existente depende fuertemente del hardware y los técnicos. Así, a medida que cambien los técnicos, las técnicas para el uso del hardware cambiarán. Los cambios se harán como parte de la transición desde una eficiencia orientada a la máquina, hacia la filosofía de orientación al usuario que tiene Pick.



Los usuarios tendrán que saber cada vez menos sobre cómo está construido un sistema, sin por ello perder eficiencia en el uso del hardware.

El sistema Pick como una esfera

El funciomamiento interno de un sistema como Pick requiere más detalles de los que podemos aportar en estas páginas. Nos conformaremos con dar una idea de las posibilidades del sistema y de como el usuario puede utilizarlo.

Un mundo es una buena palabra para describir el entorno de usuario. Pick sería entonces como un pequeño satélite en el mercado de ordenadores. El revolotea fuera de la periferia de los sistemas operativos actuales. Brilla en su espacio, pero es relativamente poco conocido. Tiene una superficie amistosa con lenguajes y herramientas para el usuario poco experto. Bajo la superficie encontramos distintos niveles, diseñados para los usuarios y técnicos que trabajan allí. Para los primeros, existen potentes herramientas para moverse entre niveles y cambiar de sistema. Estas pueden incluso introducir al usuario en el interior de Pick, donde se encuentra la base de datos relacional conducida por un núcleo planificador de memoria virtual. Este último soporta y potencia los niveles superiores, así como también controla las aplicaciones de superficie.

Nosotros discutiremos primero el



El sistema operativo Pick fue implementado inicialmente para ordenadores IBM XT e IBM 4300.

mundo de Pick en general, luego pasaremos a describir las diferentes secciones y, finalmente, estudiaremos los trabajos internos del sistema y discutiremos su potencial. Pero antes de nada vamos a introducir la terminología usada.

Dentro del sistema Pick, todos los datos y ficheros están bajo el control del sistema operativo. El software de memoria virtual mantiene todo el contenido de los discos del sistema sin la intervención del usuario. Para entender todas las implicaciones de un sistema como éste, necesitaremos examinar un poco los términos de Pick. La analogía con una esfera puede sernos útil para definirlos.

Por ejemplo, consideremos la superficie del mundo de Pick constituida por edificios. Cada uno de ellos forma una estructura separada. Existe un fichero único denominado SYSTEM, que hace las funciones de un directorio global, con todas las direcciones de los edificios del mundo de Pick. Dentro de cada construcción existe un directorio de los ficheros que lo componen. Este directorio se denomina «Diccionario Principal». Por ahora podemos pensar que éste contiene un directorio de todas las habitaciones del edificio. Estas habitaciones representan los ficheros de Pick, y cada habitación es en realidad dos habitaciones separadas. La primera se denomina «Diccionario» y contiene información útil para que el sistema sepa lo que encontrará en la otra. La segunda habitación es el fichero de datos propiamente dicho.

La única restricción en la cantidad de información de un fichero es la capacidad física del propio disco. Esta se organiza en unidades que denominaremos «Items», y equivale a un registro de datos en cualquier otro sistema.

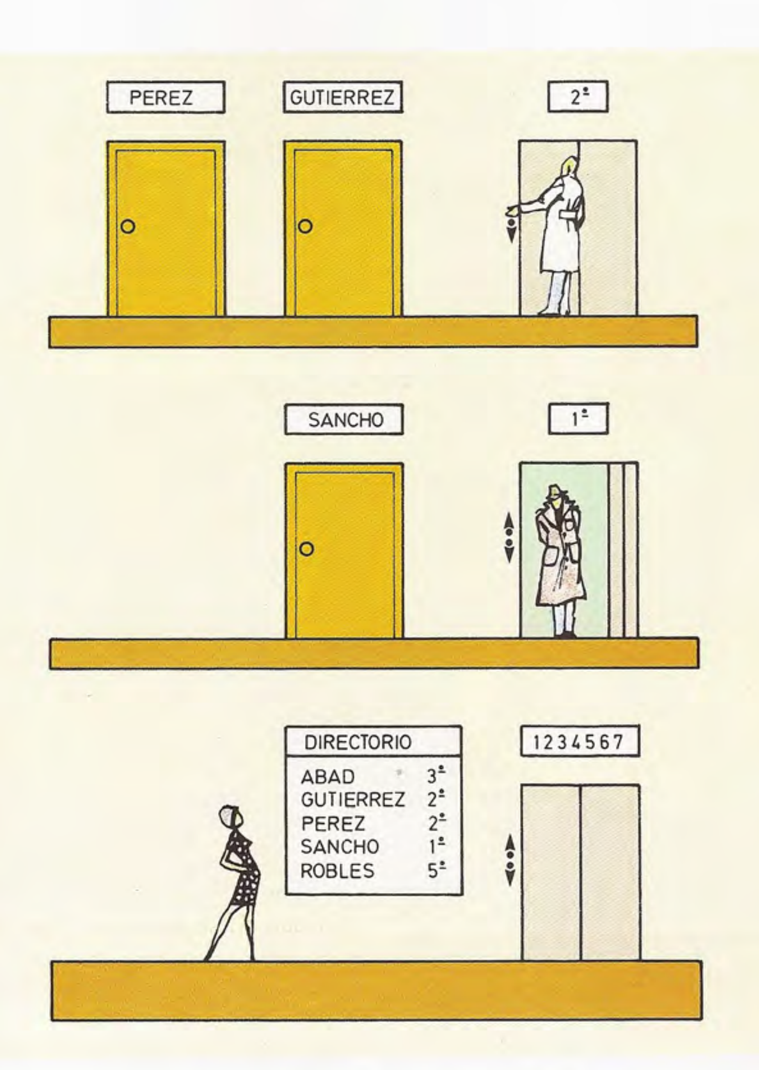
En Pick, estos «Items» tienen unas características especiales. Por ejemplo, cada uno de ellos en un fichero puede incluir datos de otros «Items» en otros ficheros. Y aún más, cada Item recibe un nombre único, Item-Id (identificador del Item), compuesto por caracteres alfanuméricos.

Dentro de cada Item encontramos varias hojas, a las que denominaremos atributo. El diccionario nos dice como leer y manejar cada atributo. Por ejemplo, cada hoja-atributo en el Item tiene que tener un procesamiento especial aplicado antes de leerse. Este procesamiento especial, frecuentemente llamado «conversión», prepara los datos para su examen y selección. Cada atributo puede ser convertido, combinado, asociado o transportado de acuerdo con las especificaciones del diccionario. Además, un atributo puede contener «valores» separados e independientes. Tanto los Items, como los atributos, valores y subvalores están definidos por caracteres especiales reconocidos por el sistema operativo Pick. Estos permiten el almacenamiento y recuperación de datos de forma automática, sin que el usuario sea consciente de ello.

Tres puntos de vista

Existen tres puntos de vista del sistema operativo Pick, todos aplicables de una forma simultánea. El primer punto de vista es el del usuario, que ya hemos visto anteriormente: un sistema de ficheros perfectamente ordenado, así como un conjunto de herramientas (Proc, Access, TCL, Basic, Editor). Para

«El mundo es una buena palabra para definir el mundo de los usuarios. El sistema Pick se puede describir cómo un pequeño satélite en el mercado de los ordenadores.»



En el sistema operativo Pick cada usuario tiene definida su área de trabajo. El fichero SYSTEM contiene información sobre las direcciones de cada zona de trabajo.

el técnico las cosas son muy diferentes. Él ve un vasto conjunto de células de memoria, numeradas de cero a diez mil. El sistema reside en estas células, y almacena y recupera datos de ellas. Se define «espacio de trabajo virtual» al conjunto de todas estas células, y su manejo se encomienda al sistema (este punto de vista también se conoce como «lógico»). El tercero y último punto de vista es el del hardware. Este nos dice cómo el sistema Pick utiliza la memoria real, los discos y cintas. Este es el punto de vista físico y está asociado con la estructura física de las unidades de almacenamiento de datos del sistema.

En lo que resta del capítulo nos limitaremos al primero de ellos.

Los procesos del usuario: TCL

Un sistema operativo se caracteriza generalmente por la forma en que la gente que trabaja con él lo ve. En nuestro caso esta interface es el denominado TCL (Terminal Control Language), el lenguaje principal que se habla aquí. TCL es la única forma de entrar en Pick y la única de salir. Todos los caminos vienen de TCL y todos los procesos acaban allí.

Uno llega a TCL entrando un identificador de usuario válido. Cada ordenador presentará su propio «logo», almacenado en un registro denominado «LOGON» en el fichero SYSTEM. Por ejemplo:

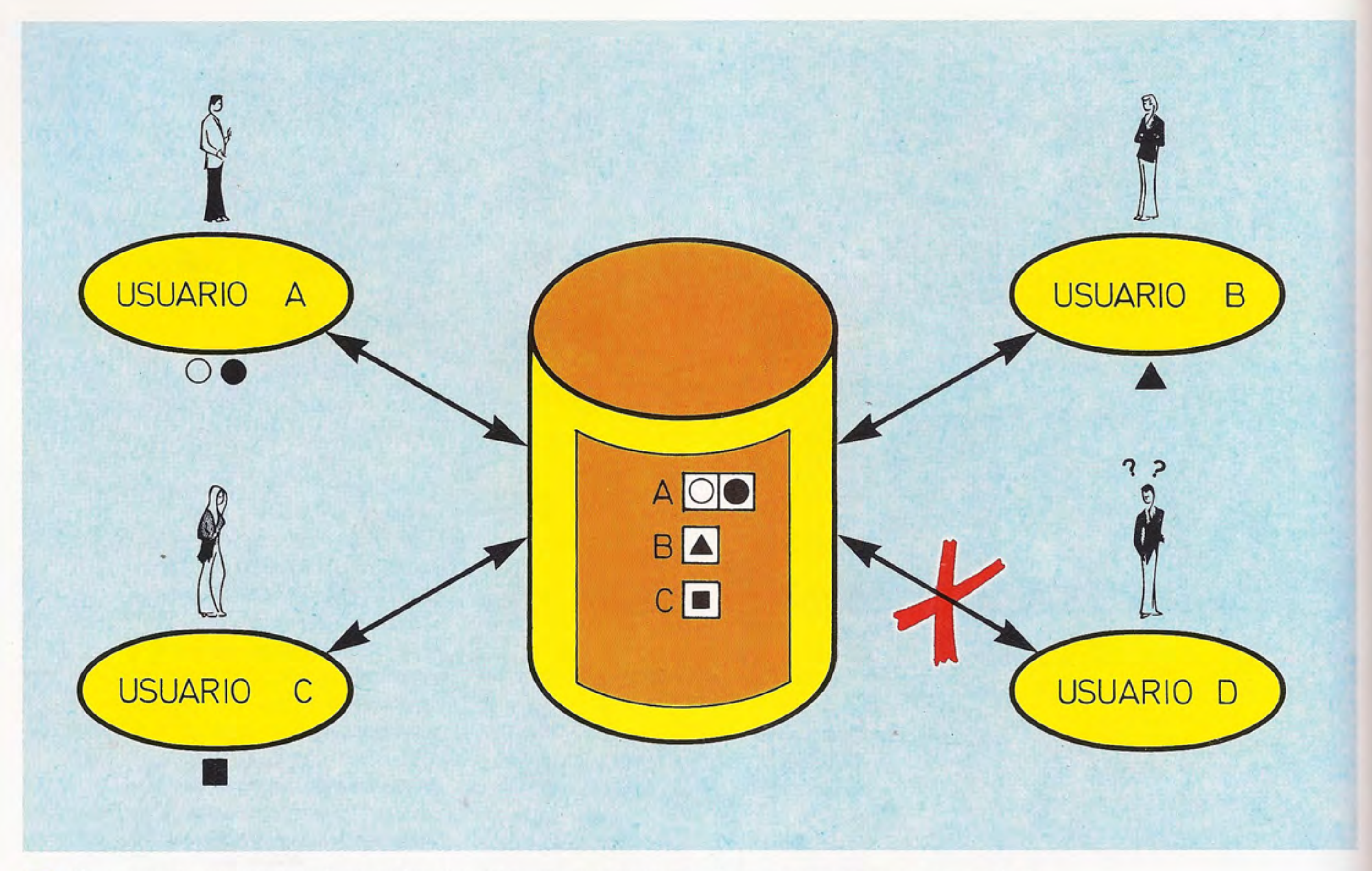
Please enter your user ID:

Un identificador de usuario es un «Item» también almacenado en el fichero SYSTEM. El identificador de usuario SYSPROG lo proporcionan todos los sistemas Pick, y es la «cuenta» usada para las operaciones de mantenimiento. Generalmente todas las entradas suelen estar protegidas con una palabra de paso.

Una vez se haya entrado, el sistema activará el «proceso de usuario», que define una determinada área de trabajo. Entonces el sistema dirige TCL al «Diccionario Principal». Allí busca un proceso con el mismo nombre que el usuario ID. Si lo localiza, éste ejecuta automáticamente los comandos allí almacenados.

A continuación el TCL muestra en pantalla el «prompt». Este varía de unos fabricantes a otros, pero generalmente

El lenguaje de control de trabajos (TCL) del sistema operativo Pick posee comandos que abren puertas a los niveles inferiores de Pick. con ellos es posible alterar las «tripas» del sistema.



Para que un usuario pueda acceder al sistema es necesario que éste posea una «cuenta», es decir, el fichero SYSTEM debe tener conocimiento de la existencia de ese usuario.

es un caracter sencillo, tal como «>». Este «prompt» significa que TCL está preparado para recibir los comandos del usuario.

TCL es una combinación rica de comandos del sistema, funciones de usuario, comandos de fichero y comandos para el control de dispositivos periféricos. Existen distintas instrucciones que abren las puertas a otros niveles de Pick. Además, hay comandos de TCL que pueden incluso alterar el sistema entero. Un usuario experimentado podría incluso crearse nuevas instrucciones y definirse un lenguaje propio combinando estas.

La entrada al sistema está siempre protegida por una palabra de paso.

En pantalla siempre aparece un prompt, que dependiendo del proceso activo será uno u otro: «>» para TCL, «.» para el editor en modo comando, «+» para el editor en modo inserción o sustitución, «*» para el depurador de Basic, «!» para el depurador del sistema.

Todos los comandos deben usarse en mayúsculas, excepto los del editor. Si cometemos algún error mientras tecleamos el comando, podemos corregirlo volviendo al lugar de la falta con «backspace» (<-) y escribiendo encima. Una vez corregido, y tras pulsar retorno de carro (<RC>), el proceso será ejecutado.

A continuación comentaremos algu-

nos caracteres especiales que modifican la forma de trabajo de TCL:

— <Ctr> X: cancela la línea actual y devuelve control a TCL. Se usa también al final de una página en una sesión multipágina para devolver el control directamente a TCL.

— <Ctr> H: Funciona de la misma manera que «backspace», se mueve el cursor a la posición anterior.

— <Ctr> R: Reescribe la línea actual.

— <Ctr> W: Mueve el cursor a la palabra anterior.

— <Ctr> J: Equivalente a «linefeed» (línea siguiente)!

— <Ctr> I: Se mueve hasta el siguiente punto de tabulación.

— <Ctr> S: Produce una señal XOFF, que para la transmisión en la línea hasta que se genera un XON o <Ctr>Q.

— <Ctr> C: Detiene la ejecución del proceso activo y devuelve control al depurador del sistema o al depurador de Basic, dependiendo del estado en que estuviese. Para volver a TCL, teclear «END<RC>». Para salir completamente del sistema, escribir «OFF<RC>. Y para entrar en el depurador pulsar G y retorno de carro.

Comandos de TCL

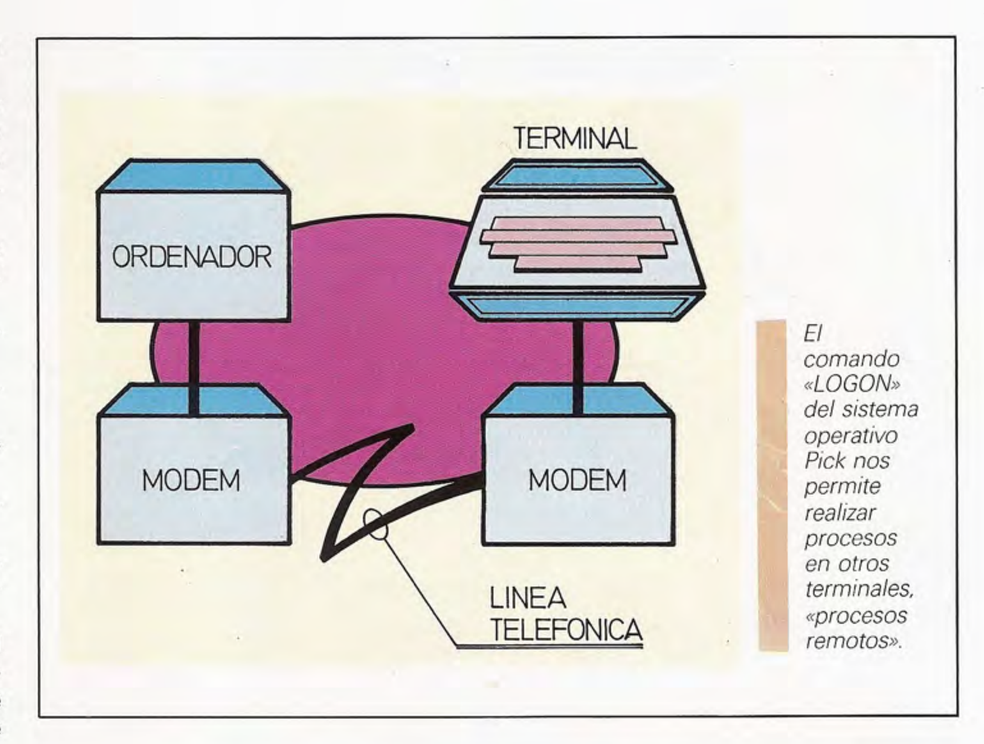
Los comandos de TCL pueden dividirse en tres grupos. El primero de ellos es TCL-I. Estos únicamente requiren una palabra seguida de retorno de carro. Alguno de ellos son triviales, tales como mostrar la hora o recordar al usuario quien es (WHO), pero otros no lo son, tales como VERIFY-SYSTEM.

En este capítulo describiremos algunos de estos comandos:

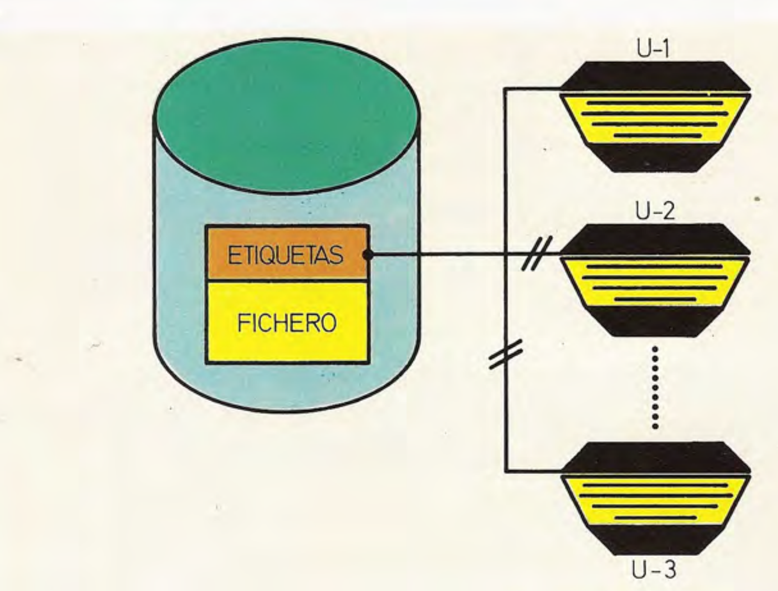
Por ejemplo, para sumar, dividir, multiplicar y restar números enteros o números hexadecimales se utilizan respectivamente los comandos: ADDD, DIVD, MULD, SUBD,ADDX, DIVX, MULX y SUBX. La línea de comando para todos ellos es la misma:

> ADDD 15 15 <rc>
30

Para realizar conversiones de hexadecimal a decimal, y viceversa, TCL proporciona dos comandos: RTD y DTR.



El comando «LOGON» del sistema operativo Pick nos permite realizar procesos en otros terminales, «procesos remotos».

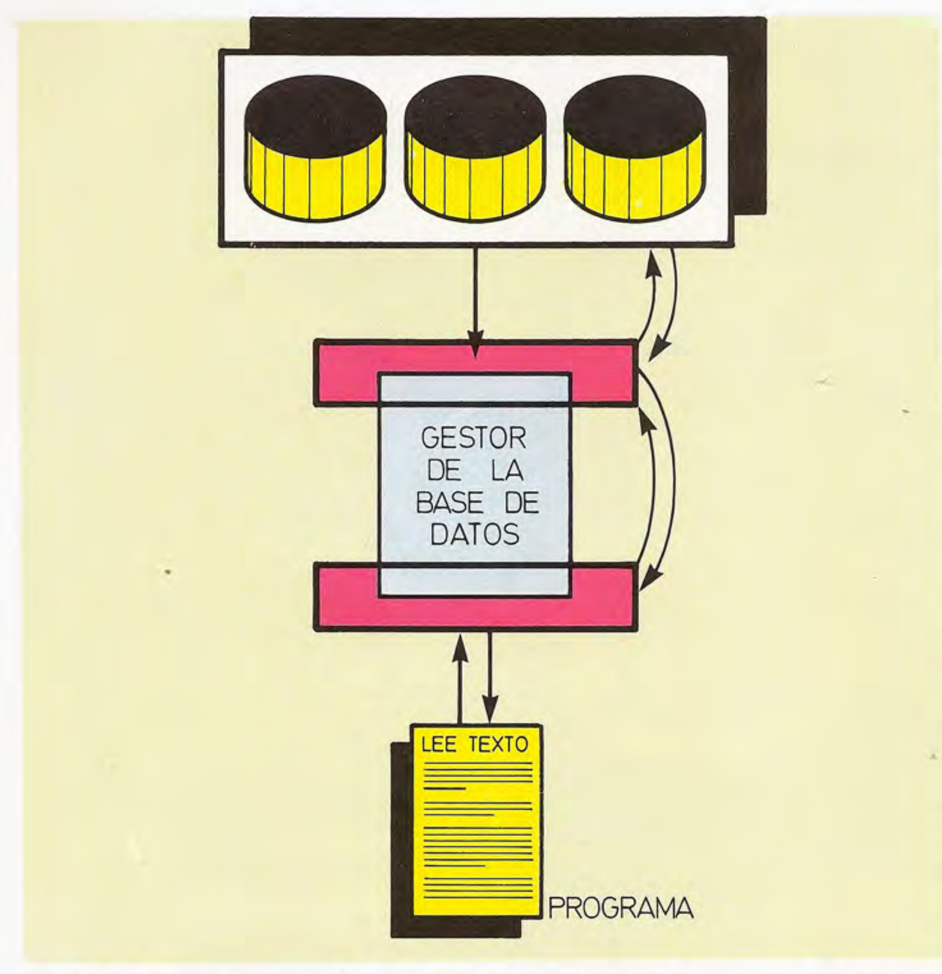


En el sistema operativo Pick los ficheros constan de dos partes. Una denominada Diccionario, que contiene los atributos del mismo, y otra propiamente de datos. En todas las operaciones que se realizan sobre ficheros es necesario especificar los dos campos.

El siguiente grupo importante de comandos son los denominados comandos «LIST», que proporcionan información sobre el propio sistema. Muchos de estos comandos permiten redirigir su salida directamente a la impresora.

Entre los más usados podemos destacar:

— LISTFILES: muestra en pantalla todos los ficheros disponibles para el usuario actual.



Para el sistema operativo Pick los usuarios no son más que conjuntos de ficheros. La gestión de los mismos se realiza como si se tratase de una «base de datos».

— LISTVERBS: muestra todos los comandos disponibles.

— LISTCONN o LISTCONN LPTR: muestra el contenido del «Diccionario Principal».

— LISTU: muestra el número del puerto, el nombre, la hora y otros datos de interés de todos los usuarios conectados al sistema en ese momento.

Otros comandos:

— LOGON: se utiliza para generar un proceso remoto en otro terminal. El comando nos pedirá el número del puerto, el nombre del usuario y, en su caso, la palabra de paso. Si la conexión se realiza con éxito aparecerá un mensaje de «LOGON SUCCESFUL». Si transcurridos unos minutos este mensaje no aparece, abortar el proceso con '<Ctr>C'.

— LOGOFF: este comando finaliza un proceso ejecutándose de forma remota en otro terminal. Al igual que en el caso anterior solicitará del usuario el número del canal y si el proceso tiene éxito se mostrará el mensaje «LOGOFF SUCCESFUL».

— El sistema operativo Pick permite el envío directo de mensajes de unos usuarios a otros. La sintaxis para este comando es: la palabra «MENSAGE» o «MSG», seguida por el nombre del usuario a quien va dirigido y todo ello seguido por el texto del mensaje. Se aconseja no usar más de 70 u 80 caracteres, para reducir la confusión en el terminal del otro usuario. si se desea mandar un mensaje a todos los usuarios conectados en ese momento al sistema, usar un «*» en el campo de «nombre del usuario a quien va dirigido el mensaje». Si no se conoce el nombre de un usuario, pero sí su número de línea, se puede usar, en vez del mismo, una exclamación seguida del citado número.

— POVF: muestra en pantalla el espacio disponible en disco.

— WHO: indica el número de línea y el nombre del usuario conectado en ese terminal. Puede usarse alternativamente para conocer quién está en otra línea: > WHO P_t (P_t representa el número de línea).

— BLOCK-PRINT: si no se le indica nada, muestra en pantalla el texto escrito. Si se indica salida por impresora con '(P)', éste lo pasará al buffer de impresión.

> BLOCK-PRINT hello (P)

Al principio de este apartado hablabamos de tres tipos de comandos en TCL. Hasta el momento hemos comentado el primer grupo. Ahora describiremos el segundo grupo o, también denominado TCL-II. En general estos comandos actuarán sobre o con los datos de un fichero o directorio de usuario. Por ello, en todos ellos será necesario suministrar al menos el nombre de un fichero o una lista de ellos. Si dicho requisito no se cumple, TCL no cederá control al proceso. Algunos de estos comandos admitirán diferentes opciones, que podrán solicitarse mediante la inclusión de la letra que lo defina (entre paréntesis y al final de la línea de comando).

Veamos algunos de ellos:

> EDIT filename itemlist options :activa el proceso EDITOR.
> COPY filename itemlist options :activa el proceso COPY.
> BASIC filename itemlist options :activa el proceso compilador de Basic.

En todos ellos, el «filename» se refiere a la sección de datos de un fichero, o a la sección de diccionario de un fichero si se usa 'DICT filename'.

>CREATE-ACCOUNT: éste es un comando que debe usarse desde el usuario SYSPROG, y sirve para «abrir cuenta» a nuevos usuarios. Durante este proceso de creación, en definitiva, lo que se hace es crear un nuevo «Diccionario Principal» y añadirlo al fichero SYSTEM. Al fichero abierto se le añaden procesos estándar contenidos en un fichero de-

nominado NEWAC, que pertenece a SYSPROG. NEWAC contiene cuatro tipos de «Items»: ficheros, comandos, Items ACCESS y procedimientos (PROCS). Durante el proceso de creación de una nueva cuenta el sistema solicitará el nombre del usuario, la palabra de paso, el nivel de protección de sus ficheros, el nivel de privilegio...

> CREATE-FILE: se utiliza para añadir un nuevo fichero a la cuenta del usuario. El único requisito exigido es que no exista ya en nuestro Diccionario Principal el nombre usado. La sintaxis de este comando puede ser de tres formas:

a) > CREATE-FILE filename modulo¨,separation } modulo¨,separation }
donde se puede construir simultáneamente la sección de datos y la de diccionario.

b) > CREATE-FILE DICT filename modulo¨,separation }
si únicamente se utiliza para la sección de diccionario.

c) > CREATE-FILE DATA filename, filename
si únicamente se quiere construir la sección de datos. El primer nombre representa el diccionario y el segundo la sección de datos.

> CLEAR-FILE: es un comando que se utiliza para vaciar un fichero, pero manteniendo su espacio. Análogamente al caso anterior, podra usarse para ficheros de diccionario o de datos.

> DELETE-FILE: tiene una función parecida al comando anterior, pero en este caso el espacio en disco es liberado.

El editor de Pick

El sistema operativo Pick tiene un único editor, denominado EDITOR. En el apartado anterior ya vimos como se le llamaba:

> EDIT filename itemlist options

Este presenta dos modos básicos de operación: el modo comandos y el modo inserción o sustitución. En el primero de ellos, el prompt usado es un punto y desde el puede llamarse cualquier comando del editor. El segundo modo normalmente se indica con un prompt '+' inmediatamente a la derecha del número de línea actual. Cualquier carácter que se escriba se tratará como dato. Para cambiar de un modo a otro bastará con pulsar I si estamos en modo comandos y deseamos pasar a inserción, o pulsar retorno de carro en caso contrario.





El sistema operativo Pick permite el envío directo de mensajes de unos usuarios a otros.

Algunos de los comandos más usados del editor son:

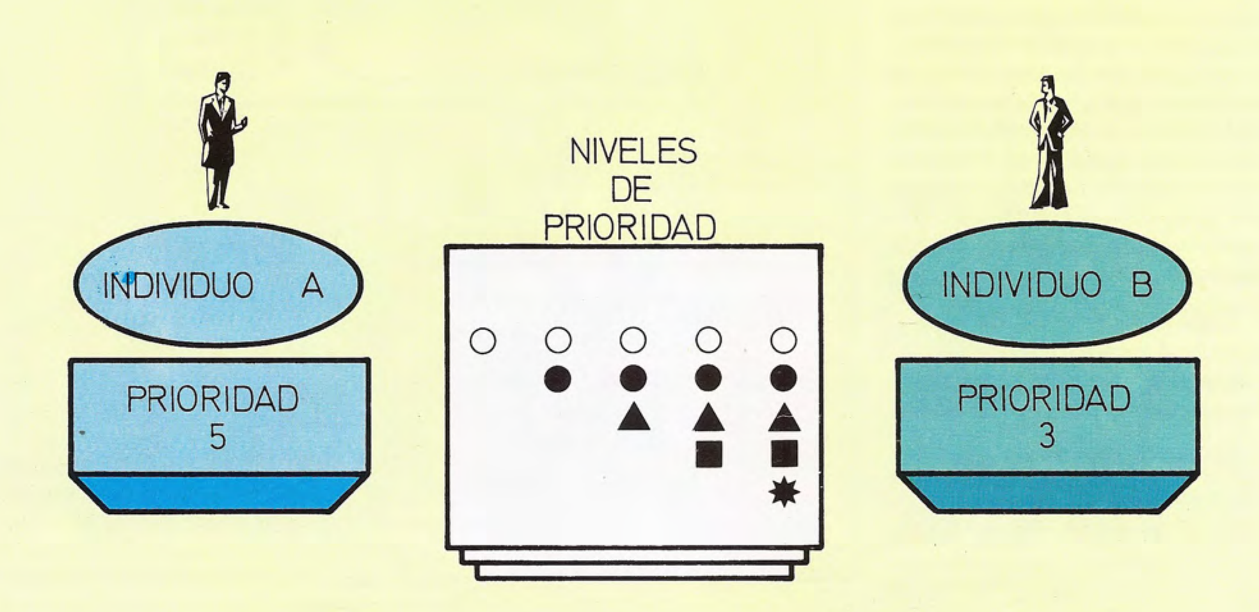
— Insertar (I): Pasa a modo de inserción hasta que se pulsa un retorno de carro cuando nos encontramos en el primer carácter de una línea.

— List (L): Realmente este comando tiene dos funciones, pero la más usual es para mostrar las líneas en el item actual.

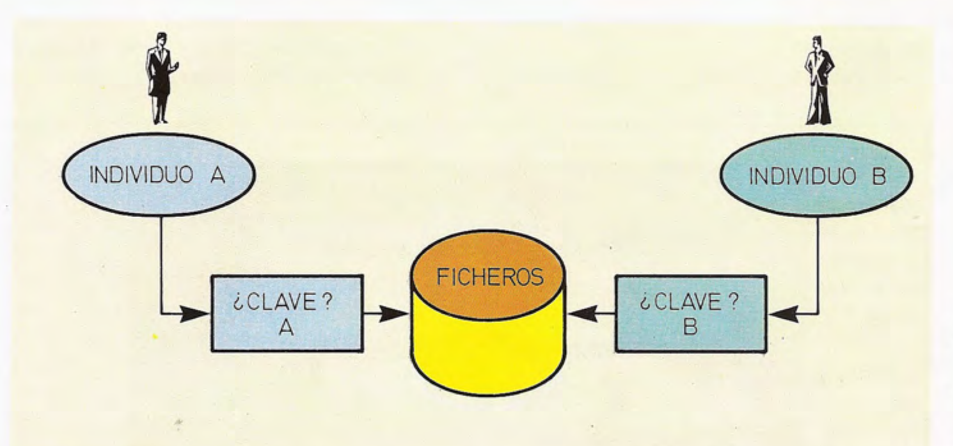
— Comando ?: Muestra el fichero y el número de línea actual.



El comando BLOCK-PRINT de TCL permite la impresión directa de texto introducido desde teclado.



No todos los usuarios del sistema Pick tienen la misma prioridad. Esta se asigna cuando se abre la «cuenta». El usuario de mayor prioridad es siempre SYSPROG.



La información que los usuarios tienen almacenada en los discos del sistema está siempre protegida. Sólo es posible acceder a ella si se conoce el nombre del usuario y su palabra de paso.

— Comando S?: Muestra el tamaño del fichero.

— Comando S: En condiciones normales el editor presenta en el margen derecho los números de las líneas. Con este comando se activa y desactiva esta opción.

— Comando G: Permite movernos a la línea cuyo número se especifica a continuación del comando.

— Comando T: Nos mueve a la línea superior del fichero.

— Comando B: El cursor pasa a la última línea del fichero.

— Comandos U y N: Mueven el cursor una línea más arriba o abajo, respectivamente.

— Comando DE: Borra la línea en curso.

— Comando FI: Abandona el editor y salva los cambios realizados.

— Comando EX: También abandona el editor, pero no salva los cambios realizados.

UCSD p-System (1)

Un sistema operativo concebido en PASCAL

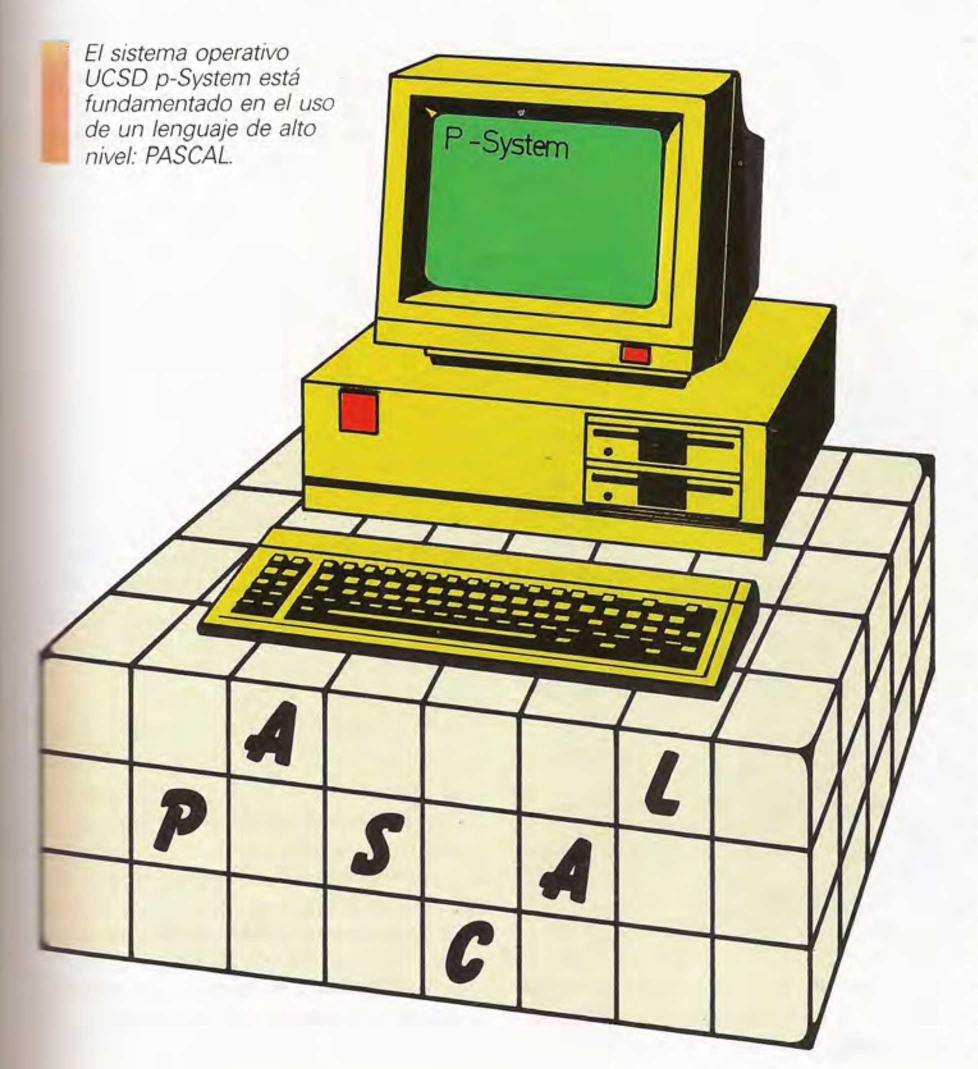


A la hora de diseñar un sistema operativo hay que hacer frente a un primer interrogante esencial: ¿se construye un sistema operativo muy optimizado, soportado por un tipo muy concreto de microprocesador y, por ende, de microordenador?... o bien, ¿se proyecta un S.O. menos rápido, pero que permita una mayor portabilidad del software a otras máquinas, al no depender en demasía de los condicionamientos hardware del sistema en el que está instalado?

La decisión por una de las dos alternativas depende claramente de los objetivos que se pretenden cubrir al desarrollar un S.O. Y no cabe afirmar, en principio, que ninguno de los enfoques sea intrínsecamente bueno o malo. De hecho, hay ejemplos de uno y otro bando en el mercado, tan importantes y renombrados como el CP/M (orientado hacia un microprocesador), Apple-DOS (orientado hacia un microordenador) y UNIX (de tendencia más universalista).

Transportabilidad ante todo

A pesar de la cuestión abierta, bien es cierto que las tendencias actuales se dirigen hacia una mayor compatibilidad y facilidad de intercambio de información entre equipos diferentes, con el fin de aprovechar las ventajas que ofrece la conexión en red de todo tipo de ordenadores.



El sistema operativo UCSD p-System está fundamentado en el uso de un lenguaje de alto nivel: PASCAL.

Entre los S.O. que siguen la consigna de la transportabilidad del software, cabe destacar el desarrollado en la Universidad Californiana de San Diego (UCSD) al final de los años 70. Este S.O., conocido como UCSD p-System, fue creado inicialmente con el propósito de implantar la programación en PASCAL dentro del entorno de los microordenadores; tendencia que se detecta fácilmente, sin más que fijarse en la cantidad de programas de utilidad disponibles como ayuda para la confección de aplicaciones en PASCAL.

La transportabilidad del UCSD p-System se fundamenta en la capacidad de lenguaje PASCAL que permite compilar o traducir programas a un código intermedio llamado pseudocódigo o código-p, situado a la mitad de camino entre el código máquina y un lenguaje de alto nivel como, por ejemplo, el FORTRAN, BASIC o el mismo PASCAL.

El código-p obtenido es el mismo para cualquier tipo de microordenador que disponga del lenguaje PASCAL. Así pues, la instalación del p-System en cualquier nuevo microordenador, sólo requiere que una parte del compilador sea sustituida, con el fin de convertir el código-p en código máquina ejecutable.

En cuanto a la zona destinada a la comunicación del sistema con el exterior, cubierta por las rutinas de entrada/salida, hay que comentar que normalmente está localizada en un módulo de código máquina llamado BIOS (Basic Input-Output System), módulo que debe ser retocado para acomodarlo a las exigencias del S.O. p-System.

La introducción de estas modificaciones en cada sistema hacen que con un mínimo cambio sea posible salvar la barrera que ofrecen los diferentes microprocesadores a la generalidad del software. Así, puede lograrse que programas desarrollados en microordenadores basados en el microprocesador 8086 de Intel, sean ejecutables sin ninguna traba en otros sistemas con distinto microprocesador, por ejemplo, el Motorola 68000.

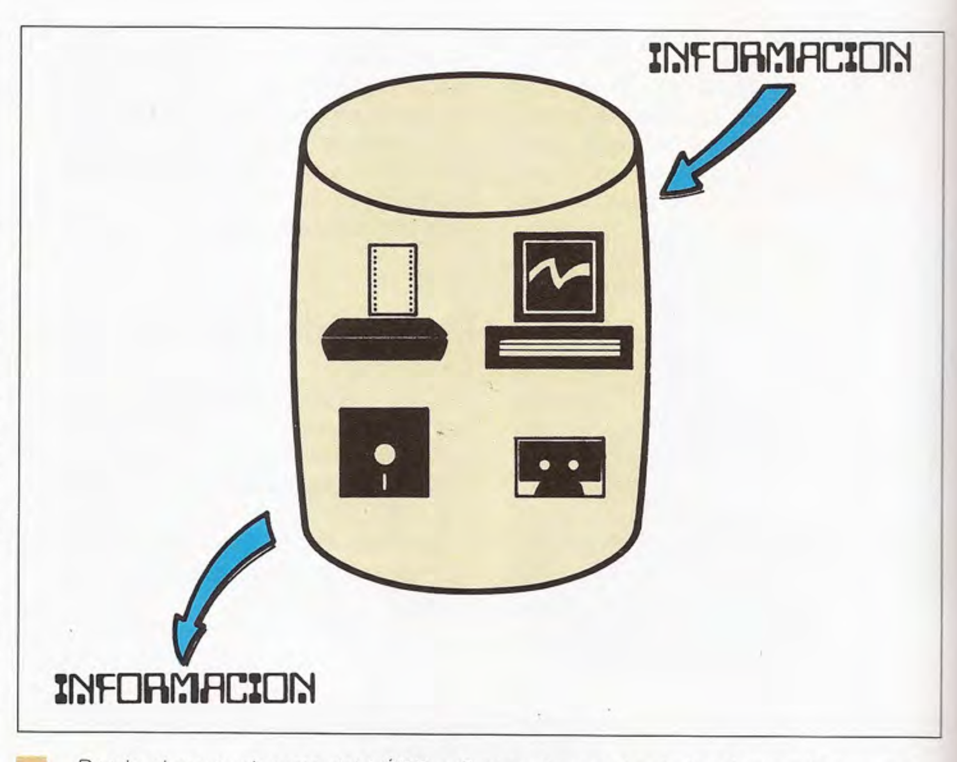
De un modo general, cuando se hace referencia a la transportabilidad de la mayoría de los sistemas operativos, se da por supuesto que se habla de los programas fuente en lenguaje de alto nivel. La versión del programa en lenguaje de alto nivel es la misma, pero ha de

convertirse en cada uno de los microordenadores que operan bajo p-System.

Elección con menús

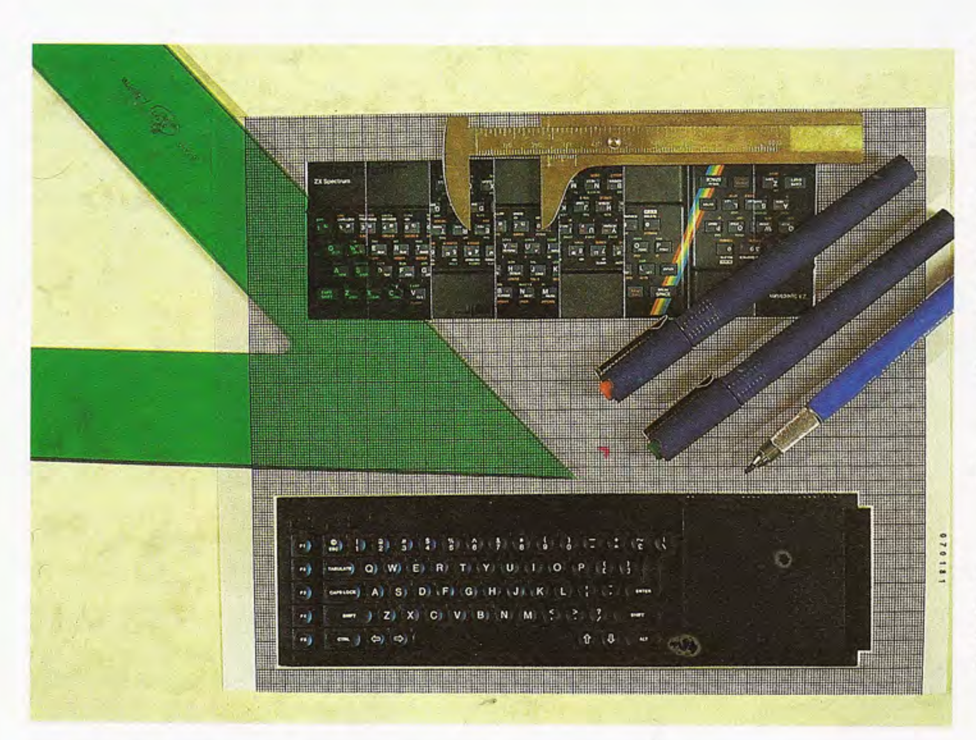
Los diversos comandos y aplicaciones disponibles bajo el UCSD p-System están organizados en menús estructurados jerárquicamente (en forma de árbol), con múltiples niveles de submenús que dan acceso a los detalles de los comandos. Este modo de funcionamiento hace que no sea necesaria una memorización de un número más o menos extenso de comandos, antes de ser capaz de actuar sobre el sistema, ya que los comandos y sus significados son visibles en todo momento.

La apariencia física de los menús difiere de la tradicional que ocupa una pantalla completa. En efecto, se asemejan realmente a mensajes del sistema, al estar indicadas todas las opciones en una sola línea de la pantalla. Ello obliga, en algunas ocasiones, a mostrar en el menú nada más que los comandos más usuales por falta de espacio. En estos casos existen submenús que complementan a los menús principales, accediéndose a ellos al responder a un menú de comandos con el signo ?.



Desde el punto de vista del UCSD p-System, la gestión de la información a través del sistema se realiza con volúmenes, o dispositivos con capacidad para almacenar, transmitir o recibir información.

Dado que todos los comandos se ejecutan al elegir opciones de menús, el usuario se ve liberado de especificar formulaciones, a veces de complicada notación. Por ello, resulta tarea fácil recorrer las diversas ramas del árbol para llevar a buen término las oportunas acciones.



El UCSD p-System está muy bien dotado de programas de utilidad que apoyan y facilitan el desarrollo de aplicaciones de la más diversa naturaleza.

Manejo de ficheros

La gestión de los ficheros con el UCSD p-System está basada en unos conceptos sencillos, con los que se construyen la arquitectura de entrada/salida del sistema.

Dichos conceptos son los siguientes:

— *Dispositivo*

Bajo este nombre se agrupan diferentes periféricos conectados a la unidad

central, tales como una consola, una impresora, un controlador de disco, etc.

— *Número de unidad*
La identificación de los distintos dispositivos conectados al sistema exige que cada uno de ellos tenga una etiqueta que lo distinga de los demás. Esto se consigue asignando a cada dispositivo un número, el llamado número unidad.

— *Volúmenes*
El sistema de ficheros de p-System está constituido por conjuntos de volúmenes. El volumen es un soporte capaz de almacenar información (en forma de ficheros) y de emitir y recibir información.

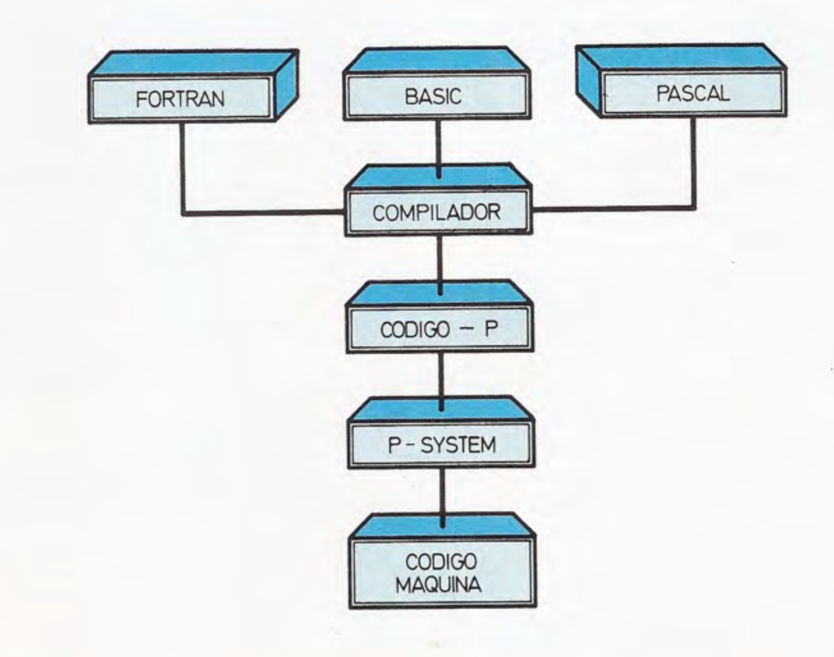
— *Nombres de volumen*
Cada volumen tiene asociado un nombre, al igual que los dispositivos tienen un número de unidad.

Todos los dispositivos, excepto los controladores de disco, son al mismo tiempo volúmenes. La impresora es un volumen que emite información hacia el exterior, la consola es un volumen que recibe información (desde el teclado) o la emite (desde la pantalla). (Bajo el punto de vista del p-System, la entrada de datos a la consola se interpreta como si procediesen de un fichero).

Los controladores de disco no son considerados como volúmenes, ya que la información no es almacenada en ellos, sino en el disco; y tampoco son capaces de emitir ni recibir información, pues su trabajo es actuar sobre los mecanismos de lectura/escritura.

Dentro de los diferentes volúmenes, existe uno que dada su importancia tiene un nombre especial (*). Nos referimos al volumen desde el cual se arranca el sistema. Este proceso de arranque consiste en copiar en la memoria interna el sistema operativo residente en disco y activar la CPU. Por lo demás, el volumen denominado *también se caracteriza por disponer de ficheros especiales utilizados para la configuración del sistema.

La referencia a los nombres de ficheros se lleva a cabo con los métodos ya clásicos, de utilización de extensiones, referencias ambiguas (wild cards), etc.



Los compiladores del UCSD p-System producen un único código, el código-p que puede traducirse a código máquina.

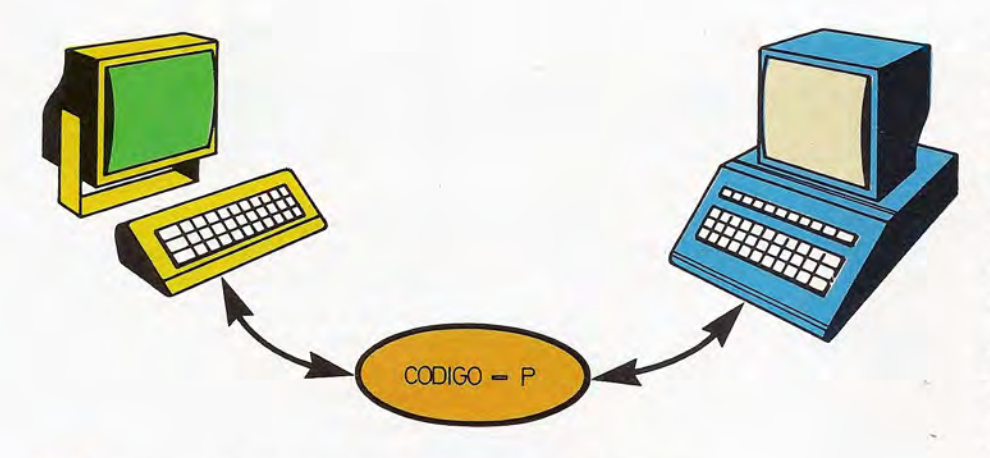


Los comandos del p-System están reflejados en menús de una sola línea, lo que facilita en gran medida su aprendizaje y uso práctico.

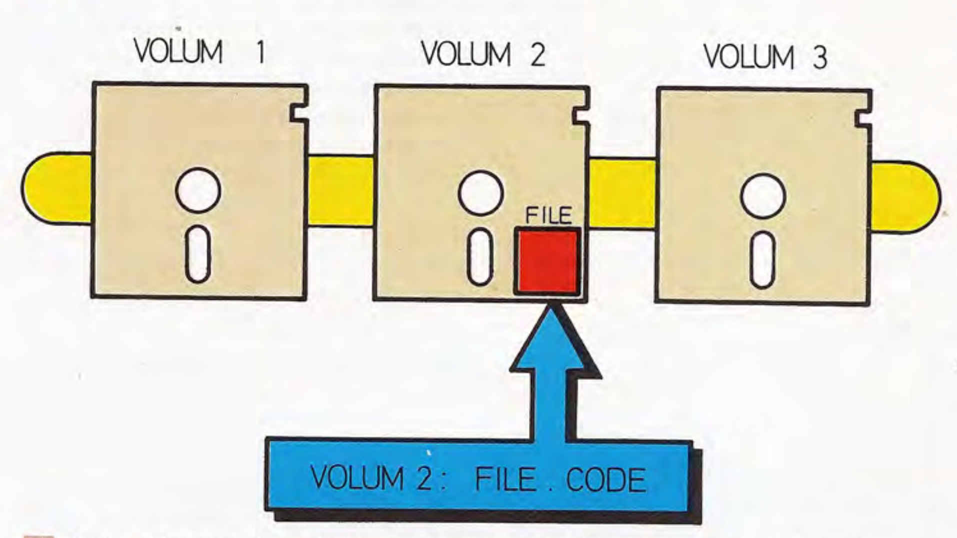
Pros y contras del UCSD p-System

En el lado positivo del p-System cabe destacar su capacidad de concurrencia (realización de varias tareas simultáneamente), la existencia de utilidades que permiten una conexión en red local con otros tipos de ordenadores, así como su fácil aprendizaje y funcionalidad.

La biblioteca de software existente es muy amplia, con lo que está garantizada la disponibilidad de una extensa colección de utilidades y programas de aplicación instalados en un rango de equipos que va desde Apple hasta sistemas multiusuario como los Digital.



La posibilidad de disponer del mismo código-p en todos los equipos gestionados por el UCSD p-System, hace que la transportabilidad de programas sea casi absoluta.



La identificación de un fichero se consigue al citar el volumen al que pertenece (VOLUM 2), el nombre del fichero (FILE), y su tipo (CODE).

La desventaja principal está en la lentitud con respecto a otros sistemas operativos más especializados y orientados hacia una máquina en particular. El probema de la falta de velocidad es consecuencia directa del modo de actuación del UCSD p-System, ya que aunque los programas son traducidos a código-p, éste es interpretado como lo son la mayoría de los programas en BASIC.

Por naturaleza, los programas interpretados son más lentos que los compilados o que los escritos en código máquina. En efecto, en la interpretación, una instrucción se traduce tantas veces como sea ejecutada dicha instrucción, mientras que en el caso de la compilación la traducción se realiza de una vez por todas.

Aún con este serio inconveniente, su velocidad es aceptable para la mayoría de las aplicaciones, debido a que el código-p es más eficiente que el BASIC y, por lo tanto, su velocidad mayor. Si las aplicaciones que se ejecutan bajo p-System son de gestión o tienen frecuentes accesos a disco, esta falta de velocidad queda aún más enmascarada. Ciertamente, el ordenador gasta la mayor parte de su tiempo aguardando entradas por teclado, u ocupándose de operaciones de entrada/salida a disco, que son varios órdenes de magnitud más lentas que las operaciones de cálculo normales.

Aunque la filosofía del UCSD p-System apunta hacia una transportabilidad total de sus programas, esto no siempre es posible; sobre todo si nos encontramos con sistemas informáticos cuyos discos estén formateados siguiendo tendencias diferentes.

Una idea de su transportabilidad y compatibilidad la da la lista de los diferentes equipos en los que es posible su instalación; entre ellos destacan miembros de las familias Apple, COMMODORE, DIGITAL, IBM, TEXAS, NCR...

Organización de la información

La organización de la información con el UCSD p-System se realiza de acuerdo a una filosofía muy simple pero verdaderamente eficaz: todo objeto que sea capaz de almacenar información (en forma de ficheros), o bien de recibirla y enviarla, es etiquetado como un volumen. El flujo de información se reduce, en definitiva, a intercambios de datos entre volúmenes.

Así pues, una impresora es un volumen que envía información al exterior del sistema, mientras que el teclado es también un volumen cuya función es la de introducir datos en el sistema.

Los disquetes se comportan igualmente como volúmenes, dado que su actividad (almacenar información) está comprendida en la definición de volumen. A cada disco se asigna un nombre de volumen con el cual puede ser referenciado unívocamente.

Para identificar la información contenida en los distintos volúmenes del sistema, deben existir ciertos convenios con los que se pueda acceder a la información por sus nombres. Así se evitará el grave inconveniente que supondría la exigencia de conocer las localizaciones exactas en donde reside la información, a través de los diferentes sectores y pistas.

Los ficheros se reconocen por su nombre y por el nombre del volumen en el cual están contenidos, actuando como separador de los dos identificativos el signo dos puntos (:).

Los diversos volúmenes se identifican



La impresora no es considerada como una entidad aparte por el UCSD p-System; sencillamente, es considerada como un volumen más sobre el que se vuelca información.

de dos formas. La primera consiste en asignar un nombre de volumen con el que éste pueda ser referenciado, mientras que la segunda se basa en la asignación de un número de unidad precedido por el signo #. Este segundo caso se diferencia del primero en que, al contrario que el nombre de volumen, el número de unidad no identifica a un solo volumen, sino que aplica el volumen residente en ese instante en el dispositivo indicado por el número unidad.

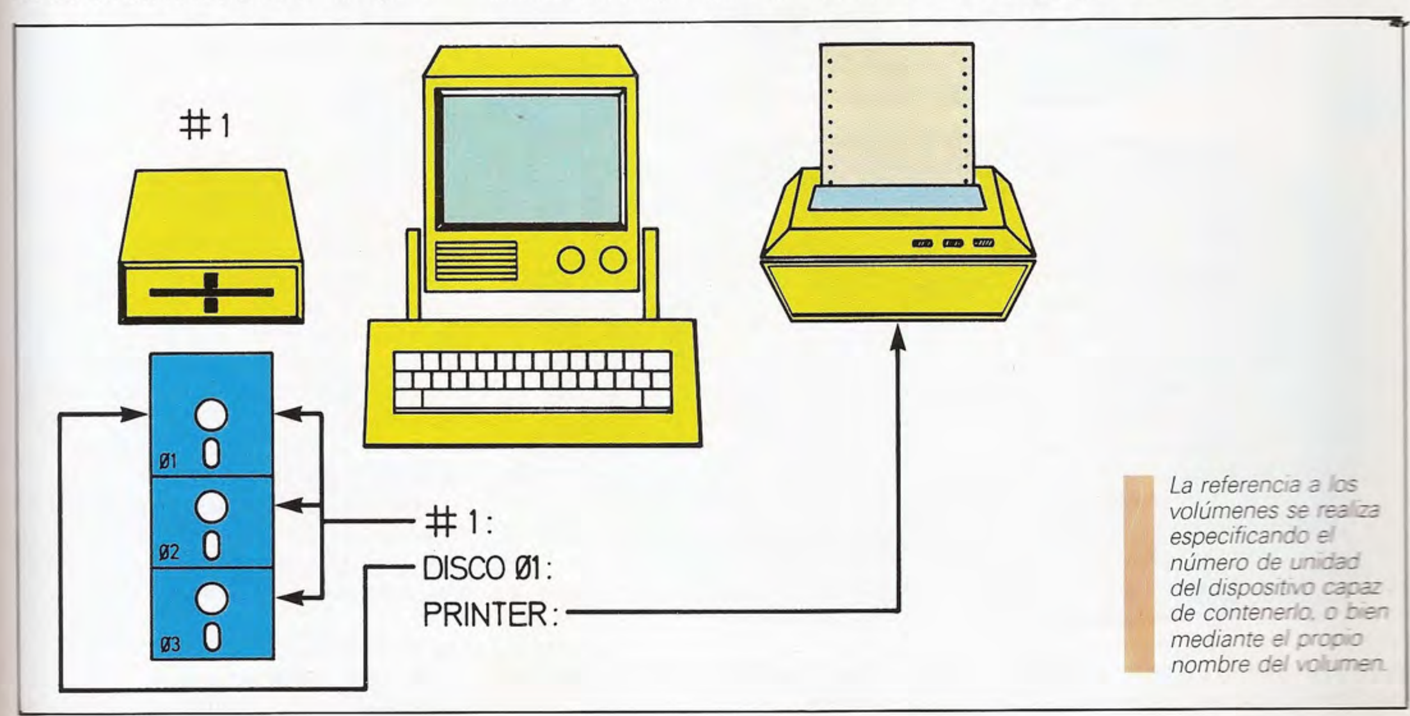
De acuerdo a estas reglas serían válidas las siguientes especificaciones de ficheros:

VENTAS : ENERO.TEXT o
4: ENERO.TEXT

Dentro de los volúmenes cabe distinguir dos tipos: el volumen de arranque o «boot volume», diferenciado por representarse con un asterisco (*) y cuyo contenido hace posible el arranque del sistema, y el volumen en curso, o «prefix volume», que es el que está activo en un momento dado. Como se ha podido comprobar en el ejemplo, los nombres de los ficheros admiten extensiones (coletilla a continuación del punto) con las que se realiza una detección más rápida del contenido de los ficheros. De igual forma, la referencia a más de un fichero a la vez es posible por medio de referencias ambiguas (wildcards).

Acceso a ficheros

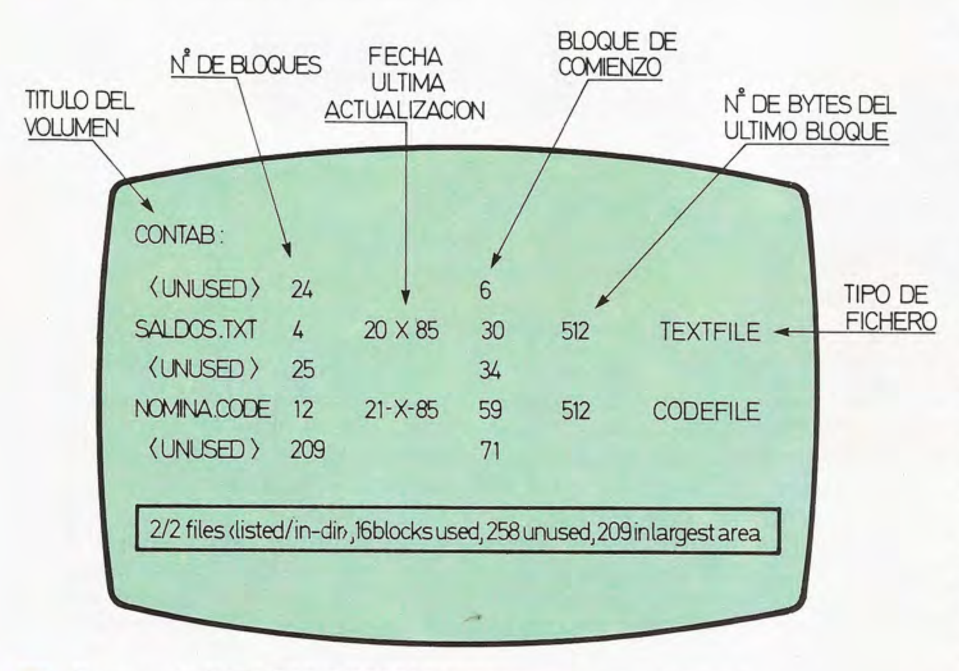
Cuando se especifica el nombre de un fichero, el sistema operativo lleva a cabo las siguientes operaciones para leer los datos contenidos en el fichero.



La referencia a los volúmenes se realiza especificando el número de unidad del dispositivo capaz de contenerlo, o bien mediante el propio nombre del volumen.

En primer lugar, el sistema operativo revisa las diferentes unidades de disco disponibles para encontrar el nombre de volumen correcto. A continuación, si se ha encontrado el volumen indicado, se consulta el directorio del mismo para comprobar que existe en él un nombre de fichero idéntico al especificado. Acto seguido, si se ha comprobado la existencia del nombre del fichero en el directorio, el sistema operativo lee la información contenida en el directorio, conociendo de este modo en qué lugares del volumen está almacenada la información del referido fichero; el propio sistema operativo se encargará de leer la información residente en las posiciones oportunas.

El sistema operativo puede regenerar ciertas zonas del disquete que no estén muy dañadas, aplicando el comando EXAMINE.



El comando EXTENDED DIRECTORY proporciona una información completa acerca del volumen en curso.

El fichero de trabajo

Dentro de los distintos ficheros que cohabitan en los soportes de memoria, cabe mencionar a uno de ellos que, si bien no es diferente de los demás en cuanto su estructura, si lo hace en cuanto a su función. El fichero al que nos referimos es el denominado fichero de trabajo, en el cual reside una copia del fichero que se ha especificado para su acceso. Su empleo resultará muy útil cuando se hagan operaciones sobre un mismo fichero, puesto que no habrá necesidad de indicar en los comandos sobre qué fichero deben ejecutarse. En efecto, si no se indica el nombre de ningún fichero, el sistema toma por defecto el fichero de trabajo.

El FILER

El tratamiento de los ficheros en UCSD p-System se efectúa bajo el monopolio de un programa, denominado FILER, que centraliza todas las operaciones a realizar en los volúmenes y en los ficheros residentes en dichos volúmenes.

Con el FILER se pueden realizar operaciones como éstas:

— Crear o borrar el fichero de trabajo.

— Cambiar el nombre, copiar y borrar ficheros.

— Copiar un fichero de un volumen a otro.

— Imprimir un fichero.

— Identificar áreas defectuosas en un disquete.

La ejecución del FILER se logra al elegir del menú principal de comandos la opción que lo identifica (F). De inmediato aparecerá otro menú que refleja los distintos comandos disponibles en el entorno del FILER. Estos son los siguientes:

- VOLUMES

Muestra los números o nombres de los volúmenes que están en línea en ese momento, así como los nombres de los volúmenes de arranque y en curso.

- PREFIX

Muestra o cambia el nombre del volumen en curso. Cuando se arranca el

sistema coinciden el volumen en curso y el volumen de arranque, siendo necesario ejecutar este comando si se desea acceder a otros volúmenes.

- EXTENDED DIRECTORY
  Muestra el contenido del directorio de un volumen. Especifica el nombre del volumen, así como los nombres de los ficheros en él contenidos, el número de bloques que ocupa cada fichero, sus fechas de creación o última modificación, el número del bloque en donde comienza cada fichero, la cantidad de bytes contenidos en el último bloque y, por último, el tipo de cada uno de los ficheros.

- LIST DIRECTORY
  Es una versión reducida del comando anterior. Muestra tan sólo los nombres de los ficheros, su tamaño en bloques y su fecha de creación o última actualización.

- GET
  Designa a un fichero como fichero de trabajo, sin necesidad de cambiar su nombre ya que basta con etiquetarlo como tal. Su utilidad se aprecia cuando se trabaja frecuentemente sobre un mismo fichero.

- WHAT
  Revela la existencia o inexistencia del fichero de trabajo; en el caso de que exista, muestra su nombre y estado.

RUN
COMPILE
EDIT
ASSEMBLE
LINK
EJEMPLO - TEXT
EJEMPLO - CODE
ARCHIVO DE TRABAJO

Cuando un fichero va a ser utilizado en múltiples ocasiones, es un buen método etiquetarlo como archivo de trabajo. De esta forma no será preciso repetir, cada vez que se escriba un comando, el nombre del fichero sobre el que se debe actuar.

Traducción e interpretación del lenguaje común

El uso de ordenadores para la resolución de problemas está condicionado en gran manera por el grado de comunicación entre el hombre y la máquina. Esta comunicación ha ido evolucionando desde las primeras y complicadas conversaciones en código máquina, hasta la llegada de los lenguajes de cuarta generación o «lenguajes naturales» existentes en la actualidad. No obstante, estos lenguajes siguen siendo poco flexibles, obligando al programador a utilizarlos de una forma muy precisa. La capacidad del ordenador para entender y actuar según instrucciones dadas en un lenguaje coloquial de uso común, tal como el castellano, está empezando a desarrollarse, aunque dista mucho de estar resuelta.

Dos son las principales dificultades inherentes al lenguaje común. La primera surge al ser el lenguaje común mucho más rico y con estructuras más complejas que las contempladas por los lenguajes que el ordenador entiende en la actualidad. En segundo lugar, el lenguaje común está evolucionando continuamente con nuevas palabras, nuevas construcciones de frases y nuevos significados para viejas palabras. Con harta frecuencia, palabras y frases tienen significados que difieren enormemente de su interpretación literal.

La complejidad del problema de la interpretación por parte del ordenador del lenguaje común, no ha sido freno en el empeño por buscar un sistema de interpretación. La clave para el futuro progreso parece ser que se encuentra en una combinación de limitaciones e interacción

En contextos limitados que incluyen mucha interacción con el usuario, es posible que el ordenador entienda suficientemente el lenguaje común como para ser capaz de responder lógicamente. Existen programas que llevan a cabo conversaciones razonablemente simples con el usuario e interpretan sus preguntas, dando las respuestas apropiadas. Están empezando a aparecer en el mercado programas conversacionales simples para diferentes aplicaciones que requieren respuestas del usuario también simples. Es previsible que en los próximos años se avance sustancialmente en el tema, debido a los recientes avances tanto lingüísticos como informáticos.



La ambigüedad del lenguaje humano hace que situaciones tan sencillas como la descrita en la figura resulten problemáticas para la máquina.

- NEW

Borra el fichero de trabajo del sistema, es decir, borra los ficheros de trabajo actuales o los ficheros SYSTEM.WRK.TEST y SYSTEM.WRK.CODE, que son los ficheros de trabajo existentes en el volumen de arranque y que son tomados por defecto al iniciar la sesión.

- SAVE

Cambia de nombre los ficheros SYSTEM, WRK, TEXT y SYSTEM.WRK. CODE, permitiendo así guardar el contenido del fichero de trabajo.

- REMOVE

Borra un nombre de fichero del directorio del volumen. La información del fichero no es borrada físicamente, sino que el espacio que ocupa es marcado como utilizable.

- CHANGE

Cambia el nombre de un fichero o de un volumen. Sustituye el nombre antiguo del fichero por el nuevo en el directorio del volumen.

- MAKE

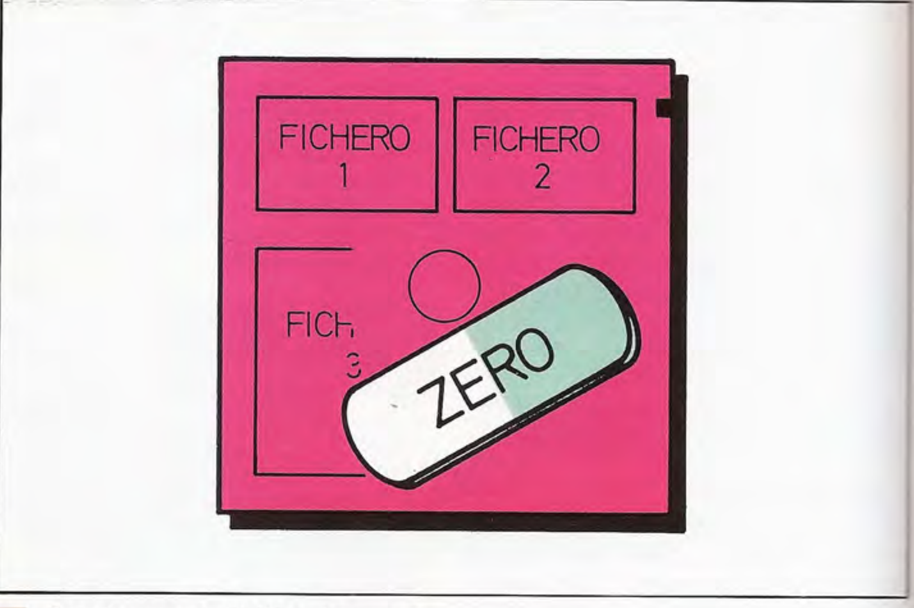
Crea un fichero al insertar un nuevo nombre de fichero en el directorio de un volumen; reserva un determinado número de bloques del volumen para la constitución del nuevo fichero.

- ZERO

Inicializa el directorio y el nombre del volumen; es decir, crea un volumen en blanco. Este paso previo es necesario para poder poblar un volumen con ficheros.

- TRANSFER

Permite mover información con las siguientes alternativas: copiar un fichero en otro volumen, imprimir un fichero, mostrar un fichero por pantalla, realizar ocpias de respaldo o de seguridad de volúmenes enteros, etc.



Con el comando ZERO se borra toda la información contenida en un volumen.

RESUMEN DE COMANDOS DEL FILER	
BAD BLOCKS	Inspecciona los bloques de un disco, identificando los defectuosos.
CHANGE	Cambia el nombre de un fichero o de un volumen.
DATE	Define la fecha del sistema.
EXAMINE	Intenta reparar los bloques defectuosos de un disco.
EXTENDED DIRECTORY	Lista los nombres de los fichereos en un volumen.
GET	Designa a un fichero como fichero de trabajo.
KRUNCH	Reordena el contenido de un volumen, agrupando espacios libres.
LIST DIRECTORY	Lista los nombres de los ficheros de un volumen.
MAKE	Crea un nuevo fichero.
NEW	Limpia el fichero de trabajo.
PREFIX	Designa un nuevo volumen como volumen-prefijo.
QUIT	Termina Filer.
REMOVE	Borra ficheros de un volumen.
SAVE	Graba con un nuevo nombre el fichero de trabajo.
TRANSFER	Copia un fichero en otro volumen o dispositivo.
VOLUMES	Lista los volúmenes existentes.
WHAT	Refleja el estado del fichero de trabajo.
ZERO	Inicializa el directorio de un volumen.

- KRUNCH

El nombre de este comando refleja perfectamente su tarea, la cual se concreta en reordenar el volumen comprimiendo los datos de manera que las zonas sin utilizar se reúnan en un solo bloque. Ello facilitará el acceso a la información.

- BAD BLOCKS

Comprueba la existencia de áreas defectuosas en el disquete, ocasionadas por el continuo uso de la superficie magnética que lo recubre.

- EXAMINE

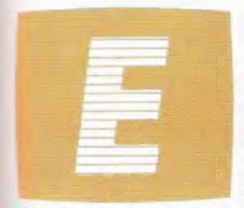
Sirve para examinar los bloques defectuosos localizados con el comando anterior. El modo de operar es el siguiente: se lee un bloque defectuoso, se escribe y, a continuación, se vuelve a escribir; esta operación se realiza un determinado número de veces, comprobándose si lo escrito es igual a lo leído.

- DATE

Fija la fecha para que sea posible marcar los ficheros con una fecha de creación o última actualización.

UCSD p-System (2)

Los tres editores del sistema operativo



En el sistema operativo UCSD p-System, el usuario cuenta con la posibilidad de elegir entre tres editores: un editor de líneas y dos modalidades de editor «full screen» o de pantalla completa. Estos editores son los siguientes:

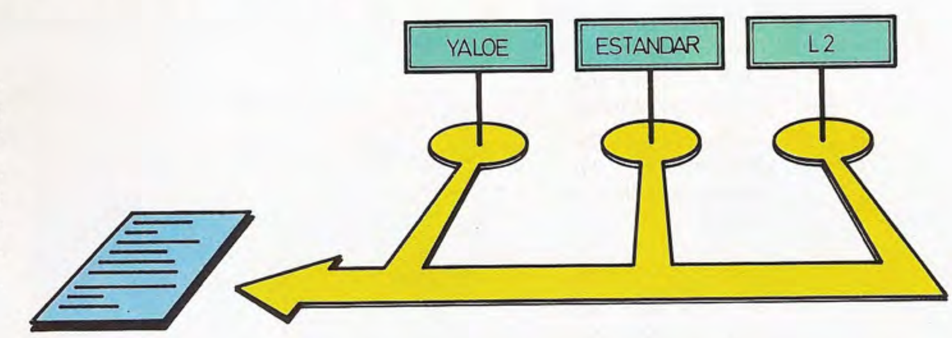
— YALOE
— Editor estándar «full screen»
— L2

YALOE esa el nombre constituido a partir de las siguientes siglas «Yet Another Line Oriented Editor». En efecto, se trata de un editor orientado a la línea, por lo que resulta interesante en aquellos casos en los cuales no se dispone de medios adecuados para la introducción de datos al ordenador. El editor estándar de pantalla completa es el que va asociado normalmente con el sistema operativo UCSD-p. Como todo editor de tipo «full screen» permite al usuario extender la actividad de edición a todo el contenido de la pantalla, lo cual exige contar con el oportuno periférico de visualización, así como con particiones de memoria adecuadas. Este editor permite la localización y sustitución de grupos de caracteres, así como la ejecución de funciones orientadas a una mayor comprensión de los programas y textos sujetos a edición.

Para su entrada en actividad, el editor estándar exige la asignación de un área de memoria (buffer de editor) en la que sea posible cargar los ficheros a tratar. El contenido de los referidos ficheros es modificado mientras reside en esa zona de memoria; luego, una vez concluida su edición, el contenido de la misma pasa a actualizar los correspondientes ficheros residentes en un soporte de memoria secundaria (en disco flexible, por ejemplo).

En este punto se hace patente una insuficiencia del editor estándar del UCSD p-System, cual es el hecho de admitir tan sólo ficheros que tengan un tamaño igual al del buffer del editor; de tal forma que no es posible cargar ficheros de superior tamaño. Esta característica desprecia su calidad respecto a los editores de otros sistemas operativos, como el CP/M o MS-DOS.

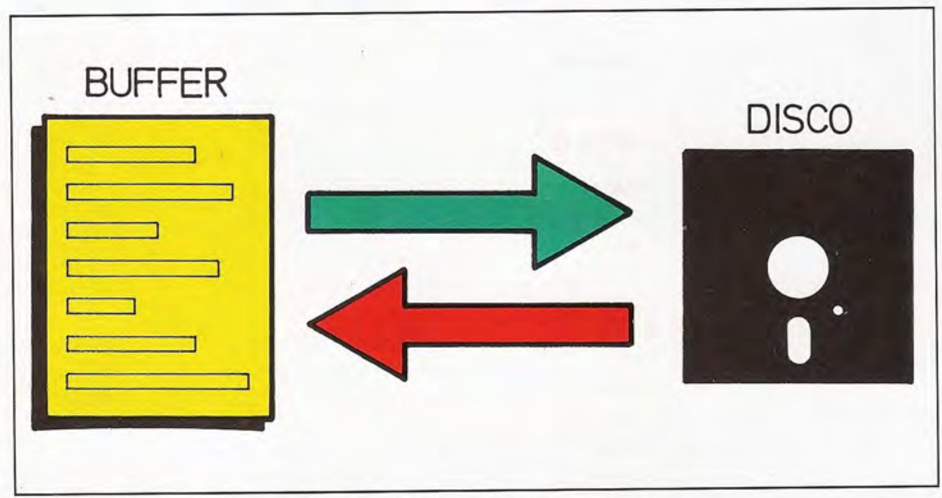
El editor L2 del sistema operativo UCSD p-System aporta una solución a las objeciones que acabamos de precisar. Mantiene igualmente un buffer de edición; pero ahora éste puede ser más reducido puesto que se mantiene un fichero auxiliar en el disco que permitirá



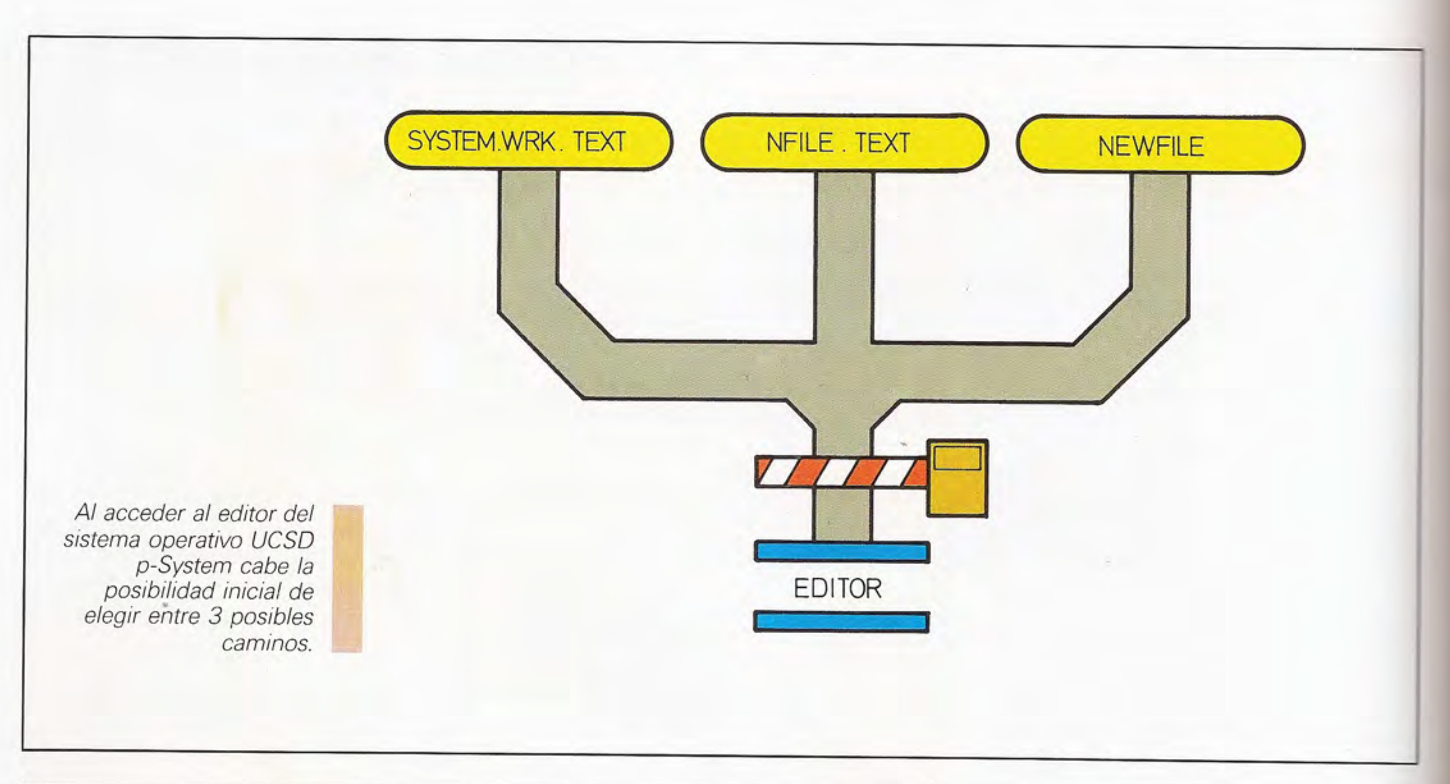
El usuario del sistema operativo UCSD p-System tiene a su disposición varias herramientas de apoyo para las tareas de edición.



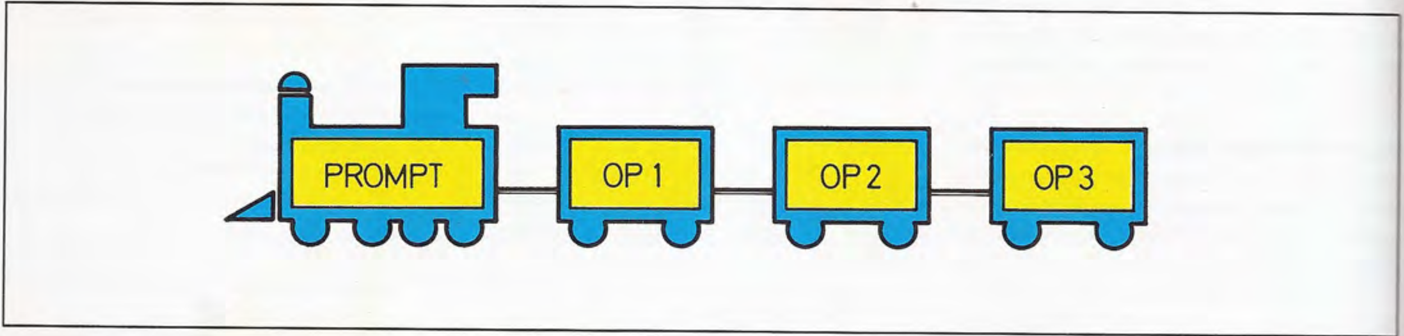
El editor YALOE de UCSD p-System está orientado a la línea.



El editor L2 actualiza el buffer de edición continuamente desde la memoria de masa.



Al acceder al editor del sistema operativo UCSD p-System cabe la posibilidad inicial de elegir entre 3 posibles caminos.



El prompt del UCSD p-System muestra la lista de opciones y comandos directamente accesibles.

ir operando con sucesivas porciones, oportunamente cargadas en el buffer de edición. Con este método no existe ya el límite impuesto por el tamaño de los ficheros a editar.

En los próximos párrafos vamos a describir las principales características del editor estándar, dado que se trata de la alternativa más común en este sistema operativo.

Como dato general, cabe señalar que éste puede operar en dos modalidades: programa y texto. En la primera de ellas se trata de ficheros que contienen programas en Pascal, mientras que en la segunda se procesan ficheros de texto en lenguaje natural, tal como ocurre con un procesador de textos convencional, aunque sin muchas de las capacidades de éste último.

Acceso al editor estándar

El acceso al editor se logra tecleando la palabra EDIT. Hecho esto, se abren tres posibilidades:

- Carga del fichero de trabajo. Sólo puede producirse si en el directorio raíz está definido un fichero de nombre:

SYSTEM. WRK. TEXT

Este proceso es automático y no se produce si el fichero no existe.

- Carga de un fichero específico. Para ello es preciso teclear el nombre del fichero sin el sufijo TEXT. Esta opción desencadenará la carga del fichero en el buffer del editor.

- Carga de un fichero en blanco. Para ello es necesario, únicamente, accionar la tecla RETURN.

Si no se ha cometido ningún error, bastará con pulsar la tecla de ESCAPE para salir del editor y pasar al modo general de comandos. En cualquier caso de los indicados aparecerá el prompt del editor, el cual contiene un resumen de los diferentes comandos que permiten controlar el funcionamiento del editor. El aspecto de este prompt es:

>Edit:A (djst C(py D(Lete F(ind I(nsrt J(mp R(place Q(uit X(chng Z(ap)

Características especiales del editor estándar

Este editor estándar del UCSD p-System ofrece una serie de posibilidades que distinguen su funcionamiento de otros editores. Cualquier usuario de un editor sabe que uno de los aspectos más tediosos, cuando es necesario efectuar correcciones, reside en la necesidad de desplazar el cursor del editor hacia atrás. Pues bien, este editor ofrece tal posibilidad de movimiento hacia atrás, lo cual se indica por la dirección que muestra el símbolo >localizado en el prompt del editor.

Pulsando la tecla < es posible cambiar el sentido de desplazamiento, con lo cual se da un toque de flexibilidad a las operaciones. Pulsando la mencionada tecla se devolverá el desplazamiento a su sentido usual.

Otra característica singular de este editor es el hecho de que permite programar el efecto de las teclas de control del cursor en forma de autorrepetición predefinida. Así, para la operación de la barra espaciadora, por ejemplo, bastará con pulsar un número cualquiera y a continuación dicha barra, para que el cursor se desplace un número de espacios igual al número introducido. Igual efecto puede conseguirse con las teclas que controlan el desplazamiento del cursor en cualquier sentido (flechas), con el tabulador, etc.

Desplazamiento del cursor

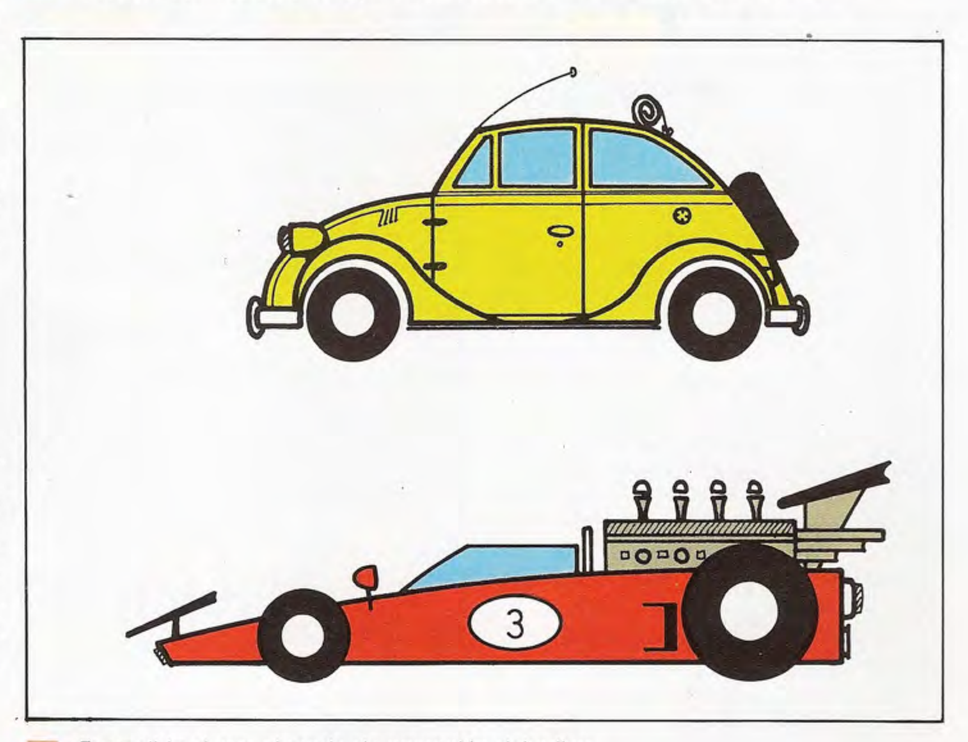
Una vez que se tiene un fichero cargado en el buffer del editor, pueden dar comienzo las operaciones sobre su contenido. Para ello es necesario desplazar el cursor a través de la pantalla, con objeto de situarlo en aquellos puntos en los que deba realizarse un proceso de edición. Los métodos de desplazamiento a disposición del usuario son los siguientes:

— Teclas de desplazamiento en cualquier dirección (flechas)
— Barra espaciadora.
— Teclas «Backspace», «RETURN» y «TAB».
— Comandos Jump y Page

La característica más relevante de las órdenes de desplazamiento es la de admitir argumentos numéricos en la forma señalada en el apartado anterior. En



El usuario del UCSD p-System cuenta con tres posibles editores para facilitar su labor de confección de programas o documentos de texto sobre la pantalla.



Es posible alterar el modo de operación del editor realizando un ajuste de sus parámetros.

todo caso, la barra espaciadora sólo admite desplazamientos en el interior de la línea en curso; aunque, igualmente acepta argumentos numéricos.

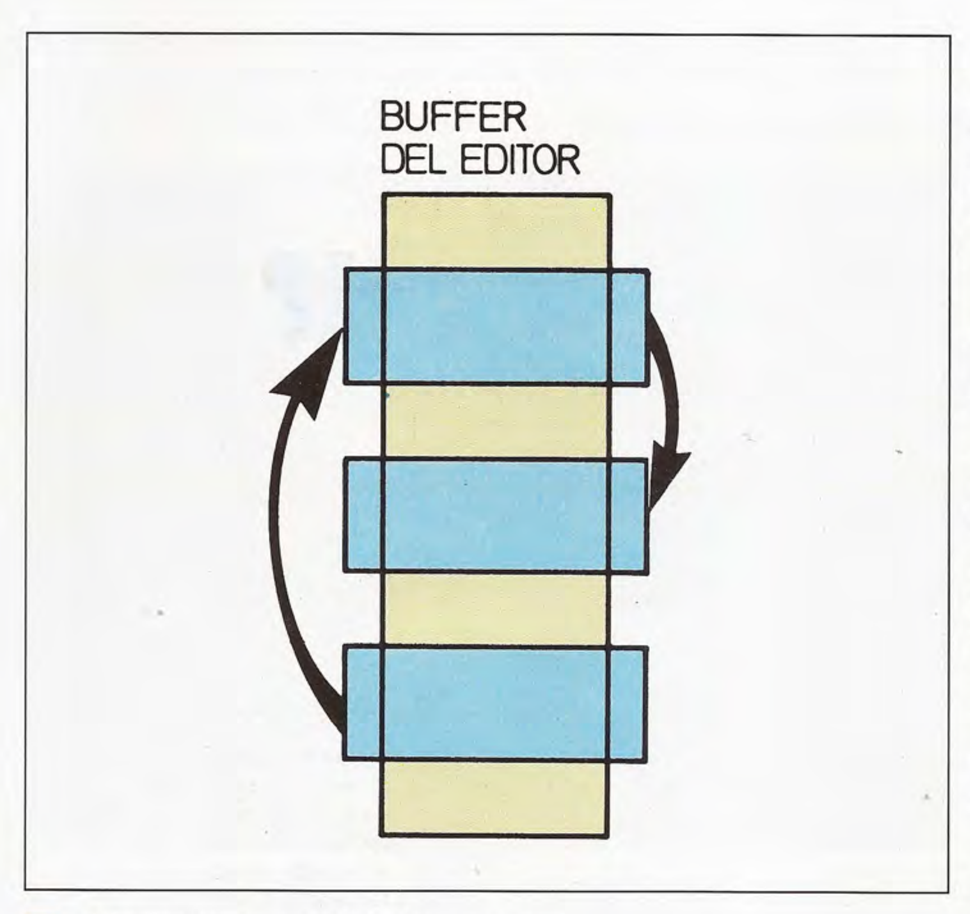
La tecla Backspace cumple una función similar a la barra espaciadora, si bien se desplaza en sentido inverso a la anterior; concretamente, su sentido de movimiento depende de los criterios fijados por el usuario.

La tecla RETURN permite saltar verticalmente de línea en línea, posicionando el cursor en el primer carácter en blanco de la siguiente. La tecla TAB hace que el cursor se desplace un determinado número de espacios hacia adelante. Por defecto este número es de 8 espacios (1, 9, 17, 25) aunque al igual que en el caso de los comandos anteriores, admite argumentos numéricos para el control del número de espacios a saltar.

El comando JUMP permite posicionarse en el buffer del editor. Su activación permite al usuario saltar con inmediatez al principio o al final de dicho buffer. Si previamente se han definido los oportunos marcadores de posición en el interior del buffer, dicho comando permitirá posicionarse sobre ellos. El salto se producirá a la marca inmediatamente posterior a la posición actual en la que se encuentra el cursor, dentro del conjunto global del buffer de edición.

El comando PAGE se emplea para gestionar las distintas páginas del buffer del editor. En su definición interna, ésta se divide en lo que se denominan páginas, que no son más que particiones de memoria de un tamaño predeterminado. Dicho buffer puede contener dos, tres, cinco o el número de páginas que correspondan por su tamaño. Por medio del comando PAGE queda a disposición del usuario un método por medio del cual es posible pasar de una página a otra sin mayores complicaciones.

Normalmente, cuando se carga en memoria el fichero que se desea tratar, la página que aparece en la pantalla es la situada en la cabecera del buffer de edición. Una vez que el usuario haya efectuado los cambios oportunos en esta página, necesitará pasar a la siguiente para proseguir las operaciones. Para ello le bastará con pulsar la tecla P y accederá a la pantalla la siguiente

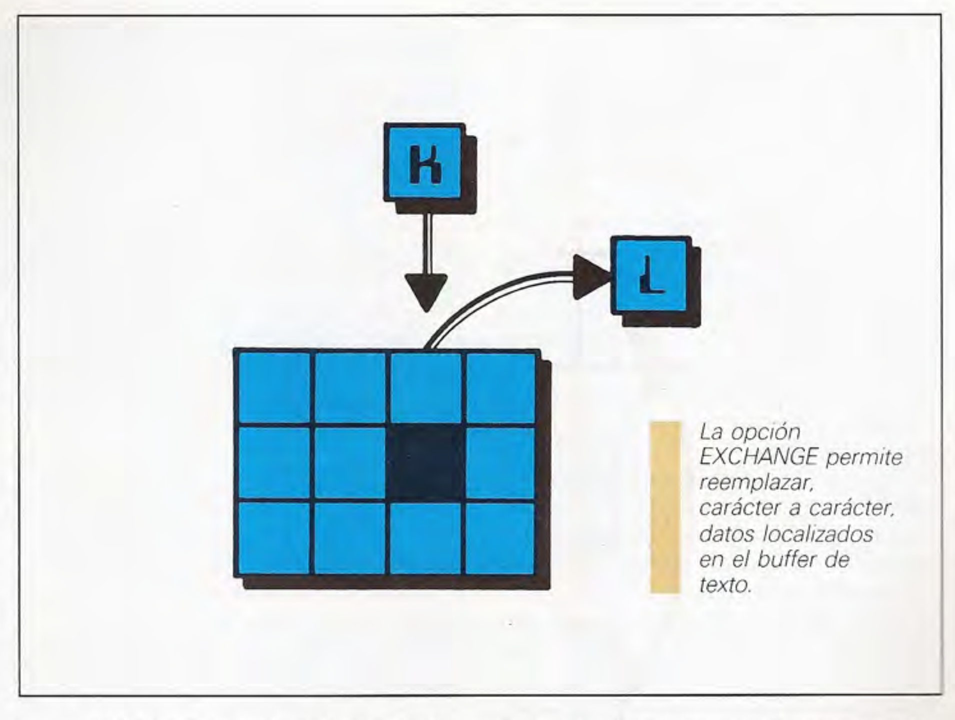


El editor estándar del UCSD p-System ofrece medios para desplazar el campo de visión sobre el buffer.



El comando INSERT permite introducir nueva información sin destruir la que anteriormente se encontraba en el buffer de edición.

página residente en memoria. También es posible ordena el retroceso a la página previa, para lo cual tan sólo hay que seguir un procedimiento similar, con la única salvedad de que el cursor que señala el desplazamiento relativo de los diferentes comandos ha de estar apuntando hacia atrás. Este tipo de desplazamiento sobre la pantalla del ordenador recibe el nombre de «scrolling» y para su control también es posible recurrir al empleo de argumentos numéricos. Si se pulsa una tecla numérica, por ejemplo la tecla 3, y a continuación la tecla P se producirá un salto de 3 páginas hacia adelante o hacia atrás, dependiendo de la orientación que posea el indicador de desplazamiento localizado en el prompt del eidtor.



La opción EXCHANGE permite reemplazar, carácter a carácter, datos localizados en el buffer de texto.

Modos de operación del editor. El comando set

El editor estándar del sistema operativo UCSD p-System puede operar, según se ha visto, en dos modos básicos: el modo programa, a través del cual es posible el tratamiento de ficheros que contienen programas en Pascal, y el modo texto a través del cual se pueden editar ficheros en lenguaje natural.

Tal selección de comportamiento se establece cambiando ciertos parámetros del editor; parámetros que están a disposición del usuario a través del comando SET por medio de sus diferentes opciones internas: autoindentación y rellenado, control de márgenes y gestión de párrafos. Normalmente, el editor se coloca automáticamente en el modo PROGRAM con objeto de controlar la operación con ficheros de programas. La oportuna modificación de los parámetros permitirá el paso el modo TEXT.

Operación del editor estándar en modo programa

El editor estándar del sistema operativo UCSD p-System está gobernado por toda una serie de parámetros, controlados de tal forma que el usuario puede modificarlos y cambiarlos según sus necesidades. Básicamente, el editor puede operar en dos modos fundamentales: el modo PROGRAM en el cual prepara ficheros conteniendo programas en Pascal, y el modo TEXT, en el que se editan textos en lenguaje natural, operando como un procesador de texto de reducidas características. Ambos modos se definen a través del comando SET, el cual permite controlar los parámetros del editor.

Cada vez que se activa el editor, éste entra por defecto en modo PROGRAM.

Dada su amplitud, no se detallarán en los próximos párrafos la totalidad de los comandos o modos de trabajo del editor a disposición del usuario, sino únicamente los más relevantes.

- INSERT

El cometido de este comando es permitir la entrada de información en el buffer del editor, sin destruir la información ya existente en el mismo. Este procedimiento puede emplearse también para introducir nuevos datos en un fichero de texto ya creado, o incluso para generar un nuevo fichero de texto. En cualquier caso, el volumen de información a introducir queda limitado por el propio tamaño del buffer de texto y de la memoria principal del sistema.

Para seleccionar la opción INSERT basta con situar el cursor en la posición en la que se desean introducir nuevos datos y pulsar la tecla I. En ese preciso instante aparecerá el prompt de INSERT sustituyendo al del editor. Dicho prompt muestra las opciones básicas a disposición del usuario. Estas son, en breve síntesis:

— Backspace: permite eliminar el último carácter incorrecto. Al activar dicha tecla, el cursor se desplaza hacia atrás y borra el último carácter.

— Delete: facilita el borrado desde el punto en el que se encuentra el cursor hasta el final de la línea previamente insertada.



En modo DELETE la información se borra sin más que desplazar el cursor sobre los caracteres a eliminar.

— Ext: pone fin al proceso de inserción y actualiza el buffer del editor.

— Escape: accionando la tecla de escape se interrumpirá el proceso de inserción, borrándose la información introducida y devolviéndose el buffer del editor a las condiciones en que se encontraba antes de iniciar el proceso.

El modo INSERT tiene además dos características especiales. En primer lugar permite operar con autoindentación (al pulsar RETURN para pasar a la línea siguiente, el cursor se situará automáticamente bajo el primer carácter no en blanco de la línea que se acaba de introducir).

Esta posibilidad es sumamente útil al escribir programas escritos en Pascal, dado que facilita la indentación del texto del programa, con las consiguientes ventajas que ello supone para la comprensión del mismo. Otra característica de INSERT se concreta en permitir el «desbordamiento» (overflow) de las líneas, escribiendo más allá del margen de la pantalla, y permitiendo por tanto líneas de mayor longitud.



El editor del sistema operativo UCSD p-System, operando en modo «program», apoya la edición de ficheros cuyo contenido sean programas en lenguaje PASCAL.

BUFFER

COPY

El comando COPY permite cargar información en el buffer leyéndola del soporte de memoria externa.

• EXCHANGE (Intercambio)

En este modo de operación, el sistema operativo UCSD p-System facilita el intercambio de caracteres del buffer del editor, uno a uno. Su operación es algo restringida, ya que sólo actúa en una sola línea y no pueden intercambiarse bloques completos de texto sino tan sólo caracteres. Para entrar en este modo de operación es preciso pulsar la tecla X. El correspondiente prompt sustituirá al del editor. Las opciones que ofrece el modo EXCHANGE son las siguientes:

— Backspace: permite retroceder un carácter, reponiendo el carácter eliminado en caso de error.

— Ext: facilita la salida de este modo de operación, actualizando el buffer del editor.

— Esc: se produce al pulsar la tecla de Escape, y origina la salida de este modo de operación sin actualizar el fichero editado y manteniendo el buffer de editor en las condiciones iniciales.

• DELETE

Como su propio nombre indica, este modo de operación permite eliminar elementos contenidos en el buffer del editor, borrando los datos carácter a carácter. Este proceso se produce despla-

zando el cursor sobre el texto a eliminar. No obstante, el texto borrado no es retirado del buffer mientras no se indique taxativamente esta opción. Para facilitar las tareas de borrado, el usuario puede modificar el sentido de desplazamiento del cursor determinado por el marcador de dirección.

La salida del modo DELETE sólo puede ordenarse en base a las opciones que siguen:

— Ext: se produce la actualización del buffer de editor.

— Esc: no se actualiza el buffer de editor, sino que éste se mantiene en las condiciones de partida antes de entrar en el proceso.

• COPY

Este comando permite copiar información de los textos que actualmente se encuentran en memoria; permite asimismo trasladar zonas de un fichero al buffer de texto. Para acceder a este comando basta con pulsar la tecla C, lo cual hará que aparezca el prompt de este comando; en él están reflejadas las diferentes opciones seleccionables.

— Esc: al pulsar la tecla de escape se abandonará el modo COPY, pasando al modo general de trabajo del editor; no será modificado el buffer del editor.

— From File: al seleccionar esta opción, lo cual se realiza pulsando la tecla F, permite cargar en el buffer de edición la totalidad o sólo una parte de un fichero residente en un periférico de almacenamiento masivo.

— Buffer: esta opción, seleccionada por medio de la tecla B, hace que se inserte la información contenida en un buffer denominado de copia, a partir del punto en que se encuentra el cursor. El buffer de copia es alimentado con caracteres cada vez que se ejecuta una acción DELETE o un comando INSERT.

• ADJUST

La misión de este comando es desplazar las líneas que constituyen el programa residente en el buffer de editor de manera que sea más legible. El desplazamiento de las líneas se logra actuando sobre las teclas de control del cursor. Otras opciones permiten el centrado de texto en la pantalla.

• Comando QUIT

Este comando pone fin a la actividad del editor, regresando el sistema al modo comando. Su ejecución admite cuatro alternativas:

— Almacenar el contenido del buffer de editor en el volumen raíz, bajo el nombre del fichero genérico SYSTEM.WRK.TEXT, el cual, si ya existe, es sustituido por el último fichero en curso.

— Almacenar los datos en cualquier volumen, dentro de un fichero cuyo nombre determine el propio usuario.

— Salida al modo general de comando sin escribir la información en ningún fichero.

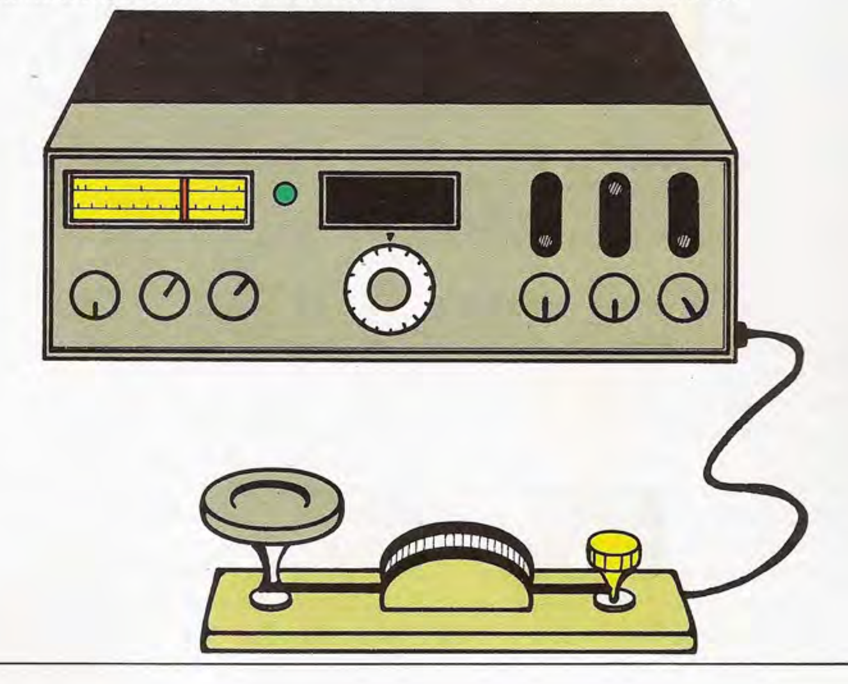
— Retorno al modo de editor sin es-

Formas de proceso de datos y su repercusión en el hardware

El tratamiento de datos admite dos variantes fundamentales cuya elección depende, esencialmente, de las necesidades del usuario del equipo informático. Tal decisión implica un desplazamiento de los requisitos técnicos del hardware que ha de soportarlo. La primera alternativa es el denominado proceso «on-line». Esta técnica permite que se realicen las transacciones y se actualicen los bancos de datos en el mismo instante en el que se lleva a cabo la operación. Un ejemplo puede ser la reserva de un billete de avión en cualquier oficina preparada para ello. Dado que en este modo de operación las respuestas a las peticiones del usuario han de ser sumamente rápidas, es necesario que la CPU sea capaz de procesar un gran número de transacciones a gran velocidad. Igualmente, se necesitan unidades de disco cuyos tiempos de acceso a la información almacenada en ellos sean los más bajos posibles. También es imprescindible que los sistemas de transmisión de información terminal-CPU y CPU-unidad de disco, sean de gran capacidad y velocidad; resultaría ilógico tener una CPU muy potente pero rodeada de sistemas de transferencia de información lentos e ineficaces.

El proceso de datos en modo «batch» supone la antítesis del proceso de datos «on-line». En él se submiten las tareas a realizar, las cuales se emplazan en una serie de colas dependiendo de sus respectivas prioridades. La CPU se ocupa de procesarlas una tras otra, con los consiguientes retrasos. Obviamente, los requisitos especificados para el proceso de datos on-line serían también deseables en una máquina orientada al proceso batch. Sin embargo, por la propia naturaleza de esta técnica, es posible rebajar las exigencias y trabajar con unidades de proceso, sistemas de comunicaciones y periféricos de almacenamiento de inferior capacidad y velodidad de proceso.

Normalmente, el proceso en modo batch es el que se emplea en operaciones tales como confección de nóminas o gestión de ficheros de personal, tareas que normalmente no requieren una respuesta instantánea, y que pueden extenderse en el tiempo.



cribir ningún fichero, con objeto de proseguir el trabajo.

Operación del editor estándar en modo TEXT

En el modo TEXT, el editor estándar del UCSD p-System opera básicamente con los mismos comandos descritos para el modo PROGRAM. Sin embargo, los parámetros que gobiernan este proceso difieren en su definición.

El control de los nuevos matices se realiza a través de la opción ENVIRONMENT asociada al comando SET.

Una vez activado el comando SET, la opción ENVIRONMENT se selecciona pulsando la tecla E. De inmediato se presenta una pantalla en la cual se visualiza el estado actual de los parámetros que gobiernan al editor, para que el usuario decida sobre su modificación. Las características seleccionables son:

— Autoindentación: cuando está en opción «TRUE», al operar en modo INSERT se producirá una indentación igual a la de la línea previa. Si es «FALSE» cada línea irá justificada a la izquierda.

— Filing: si su valor es «TRUE», cuando se salga una palabra de la línea ésta pasará automáticamente a la siguiente. En caso contrario se producirá una situación de «overflow».

— Márgenes: sólo pueden controlarse cuando la autoindentación está desactivada y el modo Filing en activo.

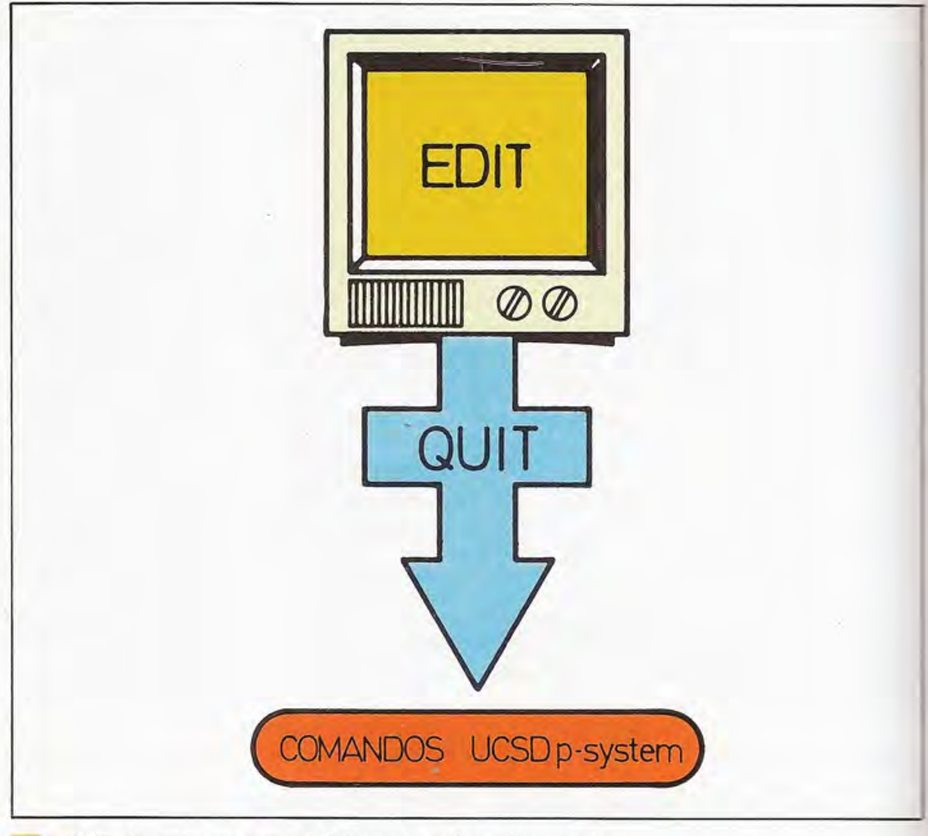
— Margen de párrafo: sólo produce indentación en la primera línea de un párrafo, con las mismas condiciones que en el caso de los márgenes.

El cambio de estos parámetros se realiza —una vez seleccionada la opción ENVIRONMENT— poniendo en práctica la secuencia de operaciones que se detalla a continuación:

— Pulsar la letra correspondiente al código de la característica a modificar.

— Si el dato a alterar es «TRUE» o «FALSE», pulsar T o F respectivamente.

— En el caso de que sea necesario introducir un número, éste debe teclearse seguido por una acción sobre la barra espaciadora. En ese preciso instante, el cursor volverá al prompt de ENVIRONMENT.



QUIT devuelve el control del sistema al nivel general de comandos, desde el entorno del editor.



Formas de almacenamiento interno de los datos

Cuando un programa está en activo, los datos que utiliza son extraídos de diferentes puntos del sistema informático; ya sea unidades de cinta, discos magnéticos o disquetes. Una vez leídos los datos del soporte externo, éstos son depositados en diversas áreas de la memoria central del ordenador; áreas que dependen de las propias características de los datos. En términos generales, estas áreas de la memoria primaria pueden catalogarse en dos tipos: zonas de Entrada/Salida y área de trabajo. Cuando un programa necesita datos que han de ser obtenidos de un fichero externo, estos son leídos y depositados en un área de entrada/salida denominada «buffer». El tamaño de esta zona de memoria primaria ha de ser, como poco, igual al tamaño del registro físico que da cobijo a los datos en el periférico en cuestión. Por ejemplo, el tamaño mínimo del buffer en el caso de leer tarjetas perforadas, se concreta en 80 caracteres (información que, normalmente, puede leerse de una sola tarjeta).

Por la misma razón, es necesario contar con la presencia de un buffer para los datos de salida. En él se irán almacenando los datos que deban enviarse a otros periféricos: unidad de disco, impresora, terminal...

El área de trabajo ocupa gran parte de la memoria primaria disponible, ya que su misión es almacenar las informaciones constantes y variables que un ordenador necesita para la ejecución del programa en curso. Así, en el área de trabajo se almacenarán datos constantes, como pueden ser encabezamientos de informes, tablas numéricas, códigos postales. Una subárea dentro de esta entidad de almacenamiento es la encargada de contener todas aquellas informaciones variables que se generan en el transcurso de un programa; por ejemplo: variables que actúan como contadores, acumuladores o, simplemente, como variables de operación.

En resumen, las áreas de entrada y salida actúan como zonas de almacenamiento intermedio de la información que fluye entre la unidad central y los dispositivos periféricos, mientras que el área de trabajo se ocupa de memorizar los datos internos implicados en la ejecución de un programa.

UNIX (1)

En busca de un sistema operativo estándar



El UNIX es un sistema operativo multiusuario (soporta varios usuarios actuando al mismo tiempo sobre el ordenador) y multitarea (ejecución de más de un programa a la vez) diseñado inicialmente para miniordenadores. No obstante, debido a su facilidad de uso, independencia del hardware y adaptabilidad a las exigencias de cada usuario, el UNIX está penetrando cada vez más en el mundo de los microordenadores.

Un poco de historia

El sistema operativo UNIX se gestó a finales de los años sesenta en los laboratorios de la Bell Telephone, pertenecientes a las compañías norteamericanas American Telephone and Telegraph y Western Electric, de reconocido prestigio en el mundo científico.

Un programador llamado Ken Thompson, no muy satisfecho con el sistema operativo de que disponía (Multics) —aunque en aquellos años éste era uno de los primeros sistemas operativos interactivos, arrastraba bastantes secuelas de sus antepasados orientados a procesos por lotes—, decidió escribir su propio sitema operativo.

Inicialmente, el UNIX fue escrito en el lenguaje ensamblador del miniordenador PDP-7de Digital Equipment Corporation. En el año 1971, el UNIX fue trasladado al más famoso ordenador de la gama PDP, el PDP-11, reescribiendo parte del sistema operativo en un nuevo lenguaje, el B (precursor del actual lenguaje C).

Ese mismo año llegó la versión del UNIX a otro programador de los laboratorios Bell, Dennis Ritchi, padre del lenguaje C, que junto con Ken Thompson tradujo enteramente el UNIX a este lenguaje.

Dada la imposibilidad de su comercialización por parte de la AT&T, esta firma decidió distribuirlo con fines puramente filantrópicos entre los Colegios y Universidades que lo solicitaran; a cambio de un pago simbólico, estas entidades recibían una cinta con el sistema operativo UNIX. Semejante decisión causó principalmente dos efectos de distinto signo. El primero fue la rápida extensión y uso del UNIX al entrar en contacto con multitudes de estudiantes y laboratorios de investigación, lo que contribuyó a que se convirtiera en uno de los sistemas operativos más conocidos dentro del mundo científico. El segundo efecto se concreta en la gran diversidad de versiones que han ido surgiendo a partir del UNIX primigenio; ello tiene su razón principal en el hecho de que no existió una única mano que dirigiese su desarrollo, así como a la gran facilidad que presenta el UNIX para recibir nuevas aplicaciones.

Para combatir este maremagnum de versiones, y aprovechando su separación de los laboratorios Bell en el año 1984, AT&T ha lanzado al mercado la versión de UNIX que pretende ser la estándar: UNIX versión V.

Las bazas del UNIX

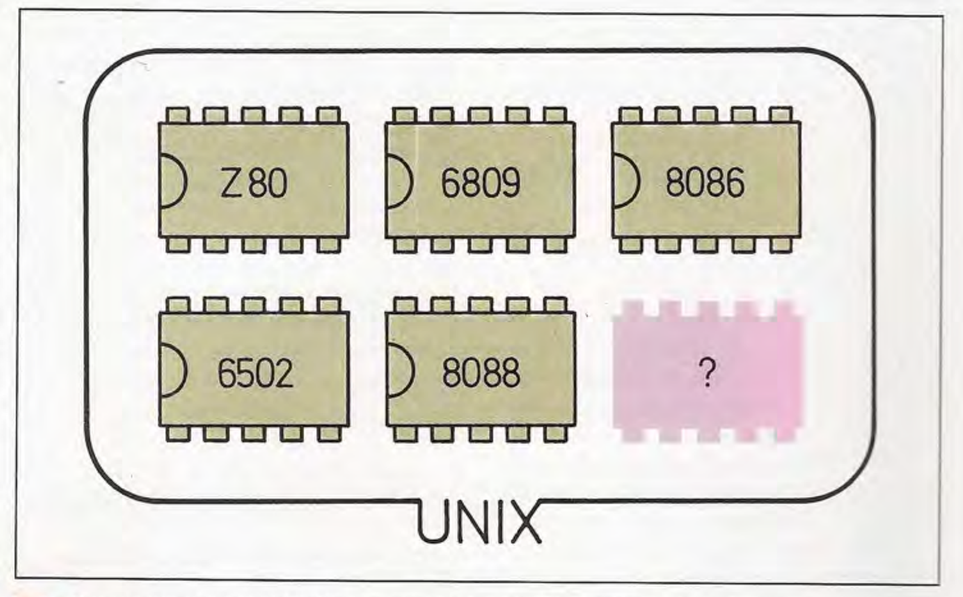
- *Transportabilidad*

La implantación del sistema operativo UNIX en cualquier ordenador queda garantizada por el hecho de que está escrito en un lenguaje de alto nivel: el C. Ello significa que no está ligado a una única familia de microprocesadores, como ocurre con los sistemas operativos escritos en ensamblador; tal es el caso del CP/M o del MS/DOS.

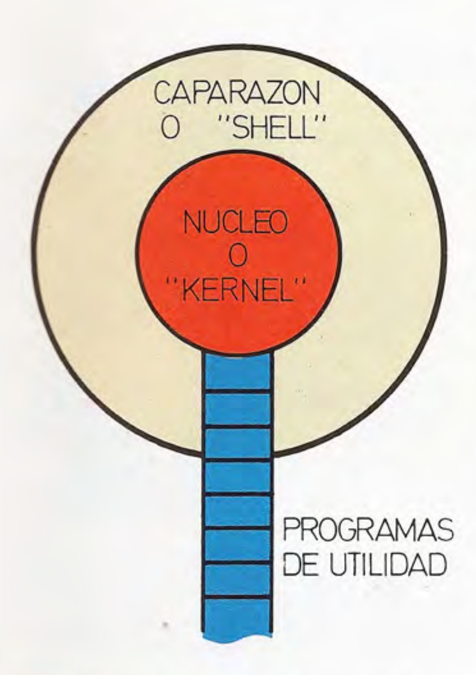
Los equipos encuadrados en el rango que va desde los microordenadores a los grandes ordenadores, están perfectamente capacitados para poder soportar el sistema operativo UNIX sin complicación alguna. Basta con un tiempo aproximado de dos o tres semanas para in-



La fácil compatibilidad del UNIX con casi cualquier tipo de ordenador, lo convierte en un firme candidato a sistema operativo estándar.



Como quiera que el UNIX es un sistema operativo escrito en lenguaje de alto nivel, resulta plenamente independiente del microprocesador que constituya la CPU del equipo.



La estructura del UNIX consta de un núcleo, encargado de realizar las tareas propias del sistema operativo, un caparazón que establece la comunicación con el usuario, y multitud de programas de utilidad.

troducir el UNIX en cualquier ordenador de nuevo cuño.

No hace falta recalcar la importancia que este hecho tiene en el actual mundo de la informática, en el que cada par de semanas aparece un nuevo modelo de ordenador, cada par de meses un nuevo chip especializado y cada par de años una nueva familia de microordenadores.

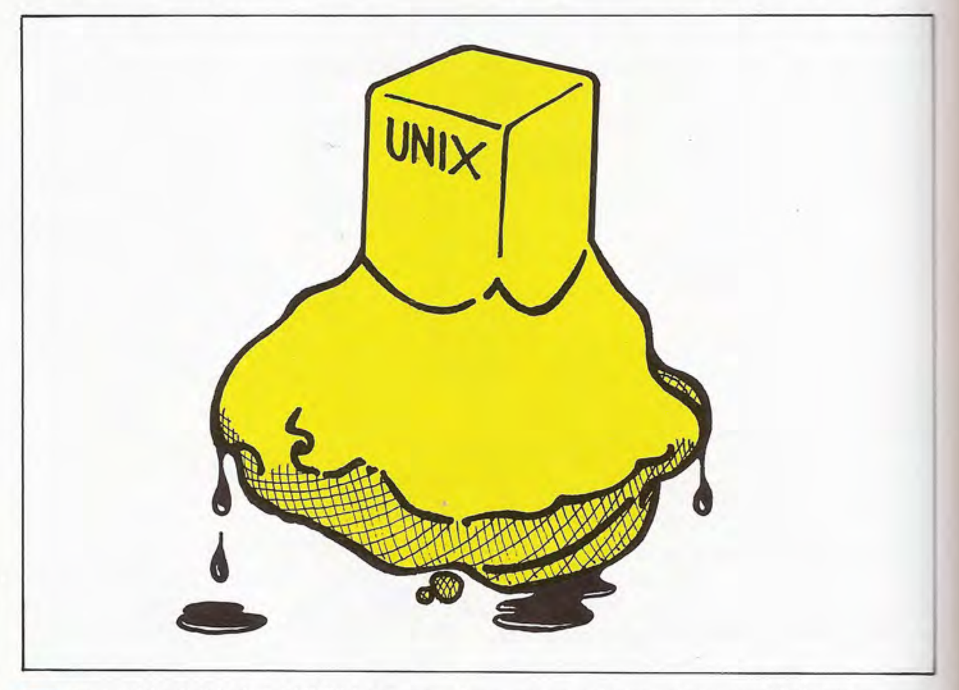
- *Software potente*

Los programas que dan cuerpo al sistema operativo UNIX están divididos funcionalmente en los siguientes grupos:

— El núcleo o «kernel»; en él residen los programas que efectúan las tareas propias del sistema operativo y gestionan el almacenamiento y recuperación de la información.

— El caparazón o «shell» encargado de reconocer e interpretar los comandos dirigidos al sistema operativo por el usuario.

— Programas de utilidad que realizan funciones de ayuda al mantenimiento del sistema y facilitan el trabajo del usuario. En el caso del UNIX la abundancia y calidad de los programas de utilidad lo convierten en uno de los sistemas operativos dotados de un mejor software de respaldo. Pueden contabilizarse hasta un total de 200 programas de utilidad; no es exagerado afirmar que raro es el programa de utilidad que necesite el usuario para el que el UNIX no ofrezca al menos dos variantes.

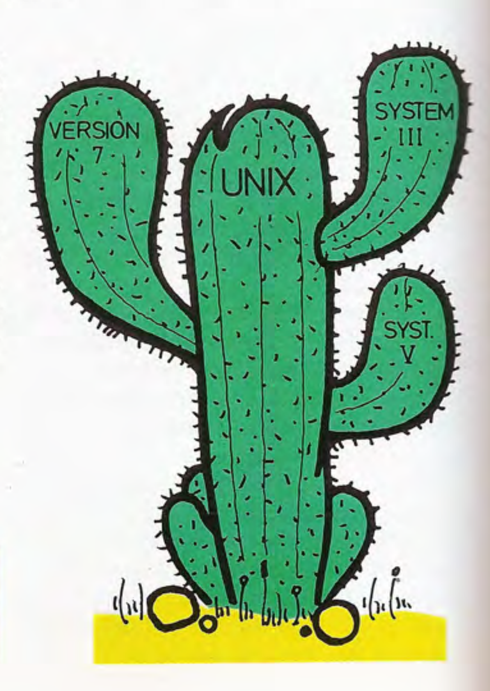


La moldeabilidad del UNIX hace que éste sea un sistema operativo fácilmente adaptable a las necesidades de cada usuario.

Así mismo, al disponer de facultades multilaterales y multiusuario, el UNIX permite desarrollar un entorno capaz de afrontar aplicaciones complejas.

- *Adaptabilidad al usuario*

Aunque un dominio completo del UNIX no se alcanza en poco tiempo, debido a la extensión de campos que abarca, sí es posible, no obstante, que un usuario que no haya entrado nunca en contacto con él sea capaz de manejarlo en un alto porcentaje de sus capacidades en un tiempo record. En efecto, el UNIX está especialmente diseñado para que la zona del sistema que comunica con el usuario, «shell», pueda ser modificada fácilmente. Ello permite acondicionar el sistema operativo a medida del usuario, estableciéndose así una comu-



La falta de una dirección única que encaminara el desarrollo del UNIX, ha provocado la existencia actual de múltiples versiones, algunas de ellas incompatibles entre sí.



Las características inherentes al UNIX permiten lograr uno de los anhelos de todo usuario: evitar que el software quede obsoleto en un corto intervalo de tiempo.

nicación amena y fácil entre éste y el sistema operativo.

• *Información estructurada*

Una de las bases más sólidas de la potencia del UNIX, reside en su estructura de ficheros. Bajo el control del UNIX todo tipo de información puede ser tratada como un fichero; ficheros que son manejados todos ellos de forma idéntica.

Como resultado de esta regularidad en el tratamiento de los ficheros, es posible el hecho de que el contenido de un fichero o de salida de un programa puedan ser enviados automáticamente a otro fichero o programa. El nombre que recibe esta conexión es «pipe» o tubería; un término que resalta muy a las claras la actividad de canalizar la información.

Volviendo a los ficheros, cabe mencionar que éstos están organizados en una estructura jerárquica en forma de árbol invertido. El elemento que ocupa la cúspide y del cual se ramifican los demás, recibe el nombre de nodo o raíz o directorio raíz. Los dos tipos de ficheros existentes en esta organización son los ficheros propiamente dichos y los directorios; estos últimos son ficheros especiales que contienen la información que precisa el sistema, relativa a todos los ficheros que se encuentran bajo su dominio.

Esta estructura está especialmente diseñada para poder identificar con sencillez a los elementos de información inmersos en un amplio marco procesable por medio del UNIX.

Algunos puntos débiles

Aunque en su conjunto el UNIX es un sistema operativo muy completo y destacable en la mayoría de sus aspectos, presenta ciertos inconvenientes, originados, principalmente, por su génesis en el campo científico y de enseñanza. Los que siguen son sus óbices más relevantes.

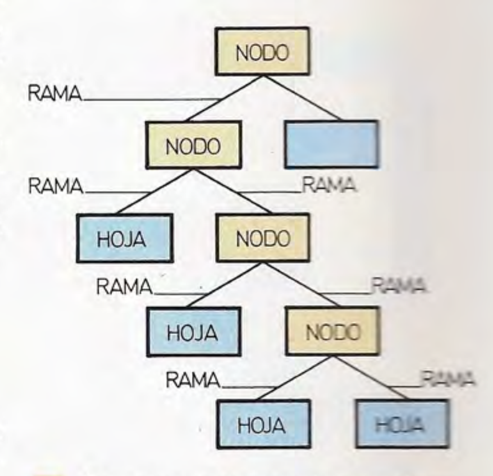
• *Proliferación de versiones*

El hecho de que el UNIX se difundiera masivamente sin estar sujeto a una compañía que centralizara su evolución y modificaciones, ha contribuido a que el desarrollo del sistema haya seguido vías muy diversas. De ahí la carencia de compatibilidad total entre sus dialectos, lo que supone un obstáculo para la aplicaión del software creado bajo distintas versiones. Cuando el UNIX llegó al mercado, AT&T disponía de varias versiones: Sexta Edición, Versión 7, PWB/UNIX, UNIX System III y UNIX System V. Para complicar más aún el panorama, la mayoría de las versiones comerciales están basadas en el UNIX System III, mientras que la firma AT&T está promoviendo el UNIX System V.

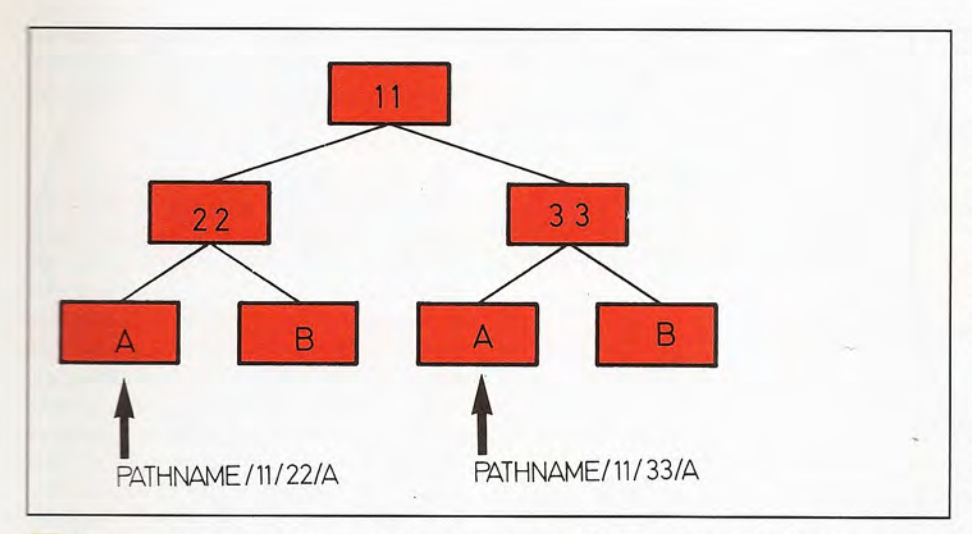
Este ejemplo es un buen reflejo del galimatías de versiones existentes. En todo caso, la reciente salida del UNIX al mercado comercial está obligando a optar por una versión específica, la System V, con la cual desarrollar el software de aplicación.

• *Relativa escasez de software comercial*

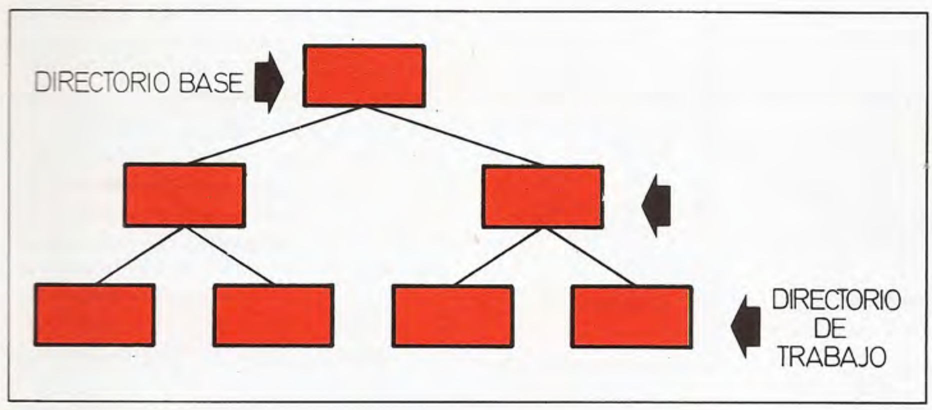
Como quiera que, en los primeros años, la mayoría de las versiones del UNIX se orientaban a la enseñanza y a



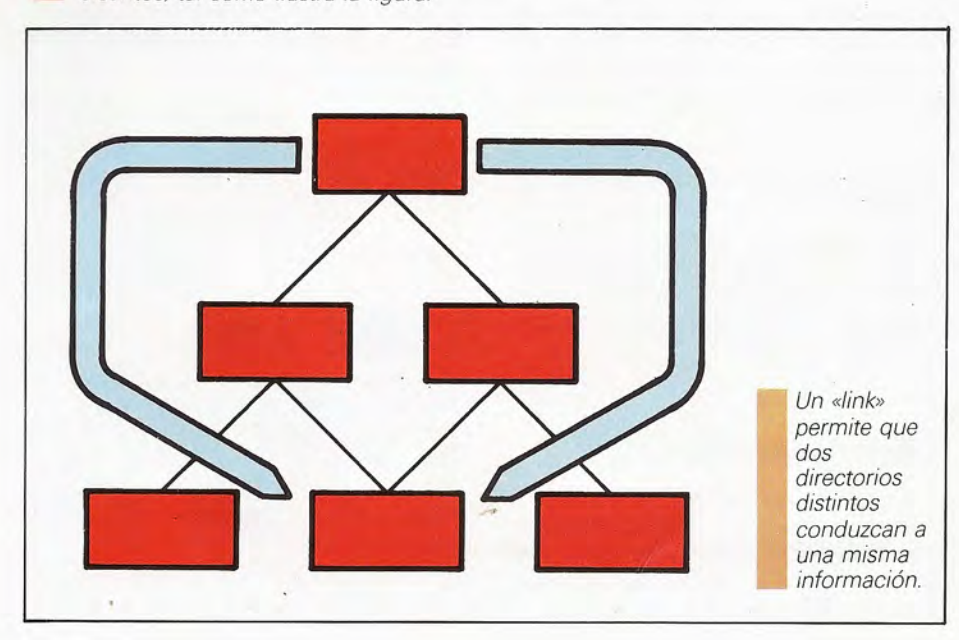
Estructura jerárquica de los archivos en UNIX y nomenclatura genérica.



Aunque coincida la denominación de los ficheros, éstos difieren bajo la perspectiva del sistema operativo, puesto que su «pathname» es distinto.



Directorio base y directorio de trabajo son conceptos distintos, tal como ilustra la figura.



Un «link» permite que dos directorios distintos conduzcan a una misma información.

las aplicaciones científicas, las aplicaciones comerciales quedaron desatendidas; principalmente, en los temas relativos a tipos de ficheros y al acceso concurrente a la información en ellos contenida, de forma que no hubiese interferencia entre los diversos usuarios. Dada esta situación, las compañías de software comercial han tenido que añadir sus propias extensiones para cubrir importantes aspectos.

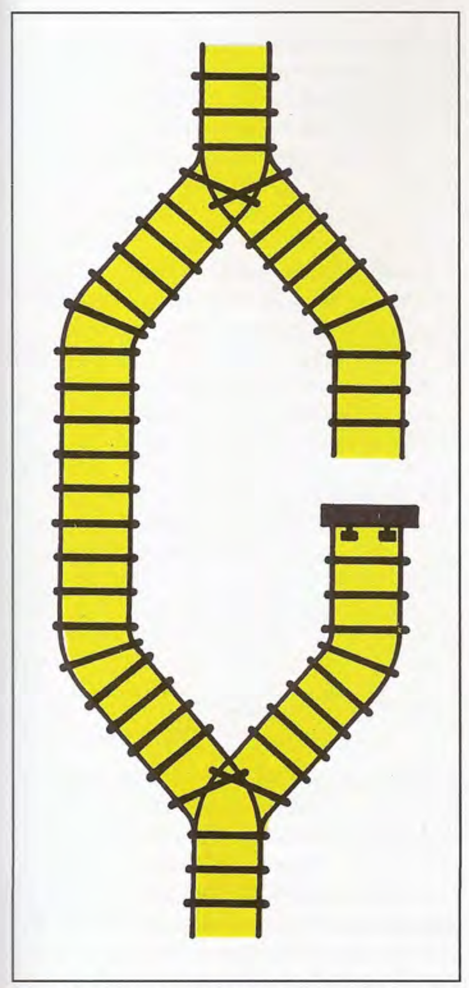
- *Consumo de la C.P.U.*

Su concepción inicial, orientada a miniordenadores, hace que el UNIX esté acostumbrado a disponer de abundantes recursos de máquina, con lo que su implantación en sistemas de reducida potencia puede saturar en cierto grado a la CPU.

La tendencia actual a utilizar microprocesadores cada vez más potentes, hace que este inconveniente se atenúe día a día.

Estructura interna y gestión de los ficheros

Todo sistema operativo ha de ser capaz de controlar una gran cantidad de ficheros y de presentar al usuario un método claro para la localización de los mismos, de manera que éste pueda acceder fácilmente a la información contenida en ellos. Para cumplir estos objetivos, el sistema operativo UNIX trabaja haciendo uso de una estructura en forma de árbol, la cual es muy común en los modernos sistemas operativos, como por ejemplo en el MS-DOS. Una estructura de este tipo comienza, en su raíz, en lo que se denomina un directorio. De este directorio empiezan a bifurcarse otros directorios, denominados subdirectorios, así como también ficheros individualizados. Esta estructura puede prolongarse indefinidamente, si bien está limitada por el espacio físico de almacenamiento disponible. Por la propia naturaleza del conjunto es posible encontrar ficheros con un mismo nombre en distintos directorios. Así, por ejemplo, imagine el caso de una empresa que almacena los datos necesarios para sus operaciones diarias en un ordenador. Dado que la empresa está dividida en varios departamentos, para ra-



La eliminación de un «link» no impide el acceso a la información afectada.

cionalizar las tareas se asigna a cada uno de ellos un directorio cuyo nombre coincide con el del departamento: «VENTAS», «PRODUCCION»... En estas condiciones es posible tener varios ficheros cuyo nombre sea «PERSONAL», conteniendo los nombres de los empleados que trabajan en cada uno de los departamentos, siempre y cuando cada uno de estos ficheros cuelguen del directorio correspondiente a su departamento.

Este esquema enlaza con lo que se denomina «pathname» en cualquier sistema operativo que ofrece esta posibilidad de ordenación. A la hora de buscar cualquier fichero, el sistema operativo sigue una trayectoria de arriba abajo a lo largo de las diferentes ramas de la estructura, pasando desde el directorio raíz por todos los diferentes subdirectorios hasta alcanzar el fichero crítico que se está buscando. Esta secuencia de búsqueda es lo que se denomina «pathname» y en tanto en cuanto sea distinta de un fichero a otro, estos serán distintos. Así, en el ejemplo anterior y suponiendo que el directorio raíz se denomina «EMPRESA», los «pathnames» de los ficheros de personal serán:

EMPRESA / VENTAS / PERSONAL EMPRESA / PRODUCCION / PERSONAL

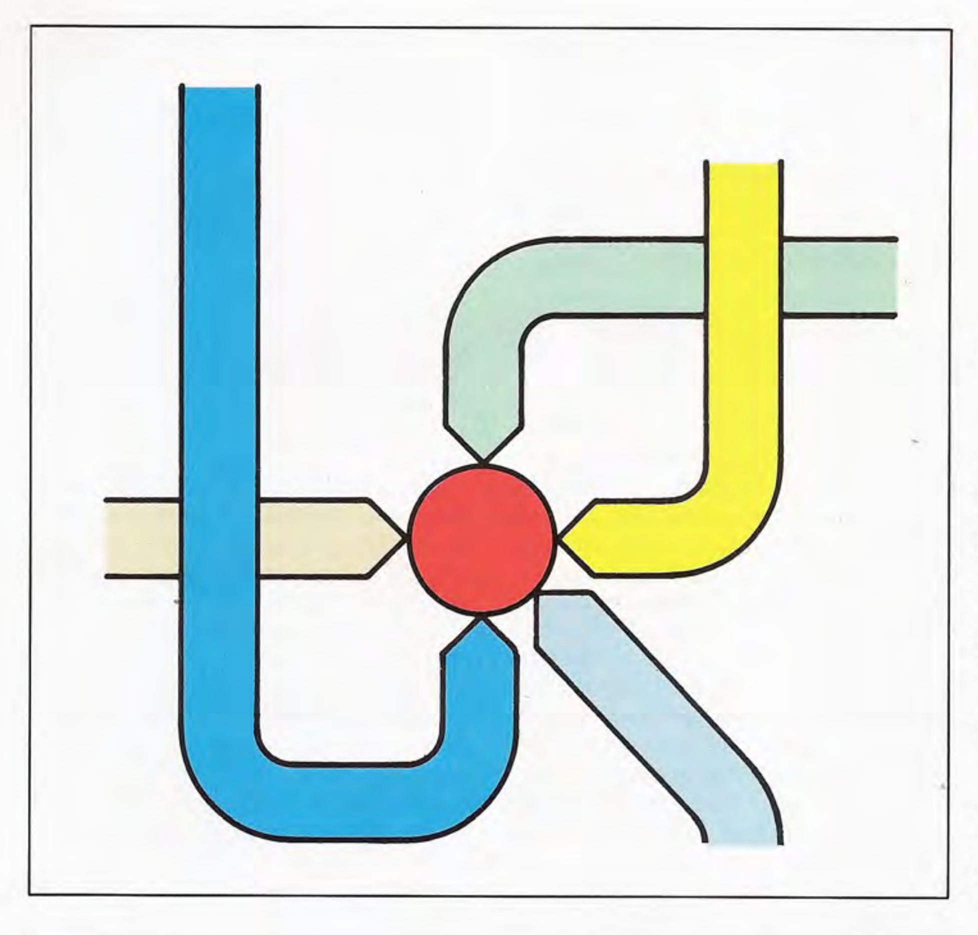
Cuando un usuario se conecta al sistema normalmente —y a través de la figura del administrador del sistema, que habrá definido previamente unos derechos de acceso así como unas áreas de trabajo permitidas—, se verá posicionado en un determinado directorio. Este directorio es el denominado directorio base, y su nombre se verá presentado en la pantalla haciendo uso del comando PWD. Si tras haber ejecutado esta orden se deseara conocer cuáles son los ficheros contenidos en el mismo, bastaría con invocar al comando IS, el cual presentaría una lista de todos los ficheros del referido directorio.

Sin embargo, existe una nueva figura entre los directorios: el directorio de trabajo. Dentro de la estructura jerárquica hay varios directorios a los que es posible acceder y en los cuales, una vez situado en ellos el usuario, para operar con la información contenida, en un fichero basta con indicar el nombre del mismo, sin necesidad de especificar el «pathname» completo. El directorio en el cual se posiciona el usuario para trabajar es el denominado directorio de trabajo o directorio «current». Si este directorio no coincide con el directorio base en el que el usuario es posicionado cuando accede al sistema, la ejecución del comando PWD mostrará el «pathname» que conduce al directorio en cuestión.

Una vez posicionado en un directorio, el usuario puede crear otro directorio, por medio de la orden MKDIR, abreviatura de la frase «make directory». Para ello debe especificar el «pathname» completo del nuevo directorio, el cual será de orden inferior al directorio de trabajo, siendo un subdirectorio del mismo, caso de especificarse así. Regresemos al ejemplo propuesto anteriormente, relativo a la estructura de la información de una empresa; suponiendo que nos encontramos en el subdirectorio «VENTAS», para crear un nuevo subdirectorio dentro del mismo de nombre «CLIENTES», bastaría con teclear la siguiente orden:



El UNIX es un sistema operativo que permite organizar la información en estructuras jerárquicas de tipo árbol.



El método de los «links» facilita el acceso a un mismo punto de la estructura a partir de distintos orígenes.

MKDIR / EMPRESA / VENTAS / CLIENTES

Una vez realizada esta operación, la ejecución del comando PWD revelará que no nos hemos movido del directorio de trabajo en el cual nos encontramos. A su vez, la ejecución del comando IS dará como resultado, suponiendo que no se hayan introducido más ficheros que los indicados en el curso de la explicación, lo siguiente:

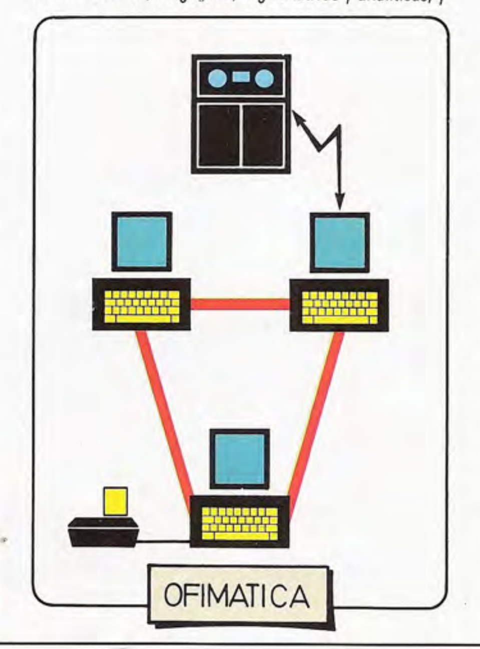
PERSONAL CLIENTES

Como puede observarse, el sistema operativo UNIX no hace ningún tipo de diferenciación, a la hora de presentar los datos, entre subdirectorios y ficheros. Otra característica de este sistema

La ofimática

El fenómeno de desplazamiento desde una economía industrial hacia una economía de la información es bien conocido en todo el mundo, especialmente en estos últimos años. Las empresas, al igual que los gobiernos, dependen cada vez en mayor grado de los sistemas utilizados para recoger, procesar, analizar y diseminar la información de forma rápida y eficaz. Estas nuevas necesidades han afectado profundamente a la filosofía de la oficina tradicional, naciendo un nuevo concepto de entender el trabajo en estos lugares: la ofimática.

Con la ofimática se mejora la transferencia de información, reemplazando a los elementos clásicos (papel, bolígrafo, máquinas de escribir, teléfonos, etc.) por equipos informáticos que aumentan la productividad gracias a la mayor velocidad y eficiencia del trabajo desarrollado. Una oficina automatizada no es —en contra de lo que cabría imaginar— un lugar repleto de máquinas, sino que debe entenderse como un entorno que permite a los empleados utilizar la interconexión de los equipos electrónicos para realizar tareas administrativas, de gestión, organizativas y analíticas; y todo ello aprovechando la facilidad ofrecida por estos equipos para el tratamiento de la información.



La ofimática está cambiando rápidamente la fisonomía de la oficina tradicional, convirtiéndola en un potente centro de información.

Con frecuencia, una compañía que cambia sus máquinas de escribir por procesadores de texto es capaz de doblar su productividad. Sin embargo, la generación de papel por métodos más rápidos y en mayores cantidades no conduce necesariamente a mejores resultados. Para que la aplicación de la ofimática resulte efectiva es necesario establecer una labor previa de formación del personal de la oficina, para que se acople a los nuevos equipos y procedimientos y sin que ello suponga un trauma.

Las futuras oficinas tendrán un aspecto muy distante a las actuales. Los terminales de ordenador serán tan familiares como hoy lo pueden ser las máquinas de escribir o los teléfonos. Estos terminales permitirán un empleo de procesadores de texto y darán acceso a bases de datos. La información, tanto gráfica como en forma de texto, será transmitida de unos puestos a otros y almacenada en soportes masivos, tales como discos magnéticos u ópticos, reemplazando a los archivadores repletos de papel. Por último, los diversos equipos estarán interconectados en redes locales y tendrán acceso a grandes redes remotas que abrirán una inmensa vía de comunicación en el mundo exterior.

operativo reside en el hecho de que para recabar información del directorio del cual depende el directorio de trabajo, basta con añadir dos puntos a continuación del comando, en la forma:

IS...

Si la información a recabar fuera la corespondiente al directorio de trabajo, este sería representado por un solo punto. Como se ha visto, el sistema operativo UNIX permite el desplazamiento por los distintos directorios, según las necesidades de la gestión informática que haya que llevar a cabo. El comando que permite llevar a cabo tal desplazamiento es CD, sinónimo de «Change Directory». Este comando asocia al usuario con un determinado directorio de trabajo, exigiendo que se especifique el «pathname» de dicho directorio en la forma siguiente:

CD / EMPRESA / VENTAS / CLIENTES

Algo a señalar y que separa en cierta forma el UNIX de otros sistemas operativos es el hecho de que cuando un usuario se conecta al sistema, su directorio base no se convierte automáticamente en su directorio de trabajo, sino que ha de ser definido como tal, caso de ser esto necesario de acuerdo con las necesidades del sistema. Al efecto se hace uso del comando CD.

Links

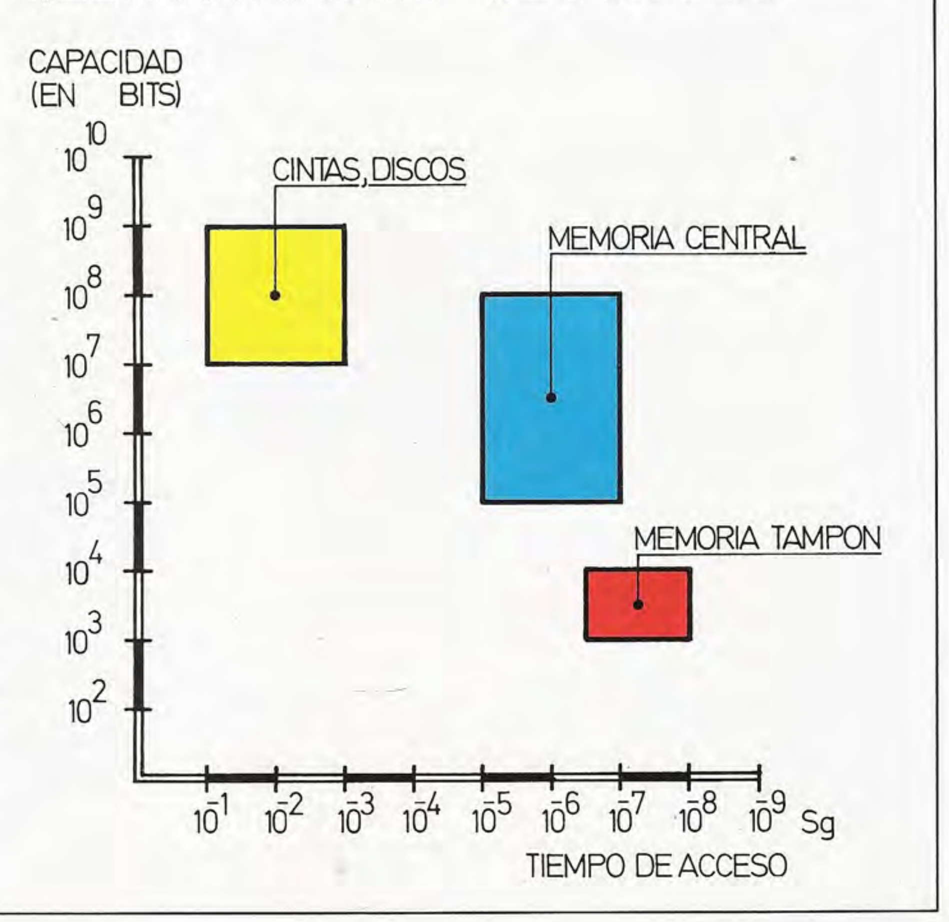
Bajo este nombre se conocen los diferentes punteros electrónicos que permiten asociar un fichero con una determinada posición en un medio de almacenamiento masivo. En estas condiciones, la especificación del nombre del fichero en un comando permite al sistema operativo acceder al mismo en el medio de almacenamiento empleado.

El sistema operativo UNIX permite, en el caso de trabajar en un entorno multiusuario, definir varios «links» sobre un mismo fichero de manera que varios usuarios pueden compartir la información contenida en el mismo, lo cual es una faceta importante para la obtención de un máximo rendimiento de los medios de almacenamiento masivo.

Cuando un fichero es creado se genera un «link» hacia él. El fichero puede ser a continuación borrado, pero esto no significa que físicamente se eliminen los registros que lo componen, sino que

Estructura jerárquica de la memoria del ordenador

El usuario de un ordenador se verá limitado a la hora de generar sus programas para el mismo por la capacidad de la memoria disponible en el equipo, toda vez que ésta es el soporte básico de la información (datos, variables, etc.) a procesar así como del propio programa. Para el usuario, lo ideal sería tener una memoria central de elevada capacidad y con una gran velocidadd de acceso; sin embargo, conseguir este objetivo sería sumamente costoso en términos económicos, así como de una complejidad tal en circuitería que anularía las posibles ventajas. Es por ello que dentro de la memoria se ha establecido una jerarquía de dos niveles: por un lado, se encuentra la memoria central, de elevada capacidad de almacenamiento y con tiempos de acceso rápidos, y por otro la memoria auxiliar de mucha mayor capacidad pero con tiempos de acceso dilatados. Dentro del primer grupo cabe diferenciar entre dos tipos. El primero coincide propiamente con la memoria central, construida, normalmente, a base de dispositivos semiconductores y cuyos tiempos de acceso oscilan entre los 250 nanosegundos y los dos microsegundos. Por otro lado se encuentran las memorias «tampón» o memorias de anotaciones, con una elevada velocidad de acceso pero escasa capacidad, y que son utilizadas en los canales de comunicaciones o en la propia CPU como memoria local; este tipo presenta tiempos de acceso variables entre los 50 y 200 nanosegundos. Dentro del grupo de las memorias auxiliares caben las memorias de almacenamiento masivo o dispositivos con acceso selectivo a la información, de elevada capacidad y con tiempos de acceso relativamente elevados. En la actualidad, el predominio en este tipo de memorias corresponde a las unidades de disco, con velocidades de acceso de aproximadamente 25 milisegundos en algunos casos. El otro apartado dentro de las memorias de este tipo lo ocupan las memorias de fichero, básicamente las unidades de cinta magnética, que por su acceso secuencial a la información se caracterizan por tener tiempos de acceso elevados.



La característica más relevante del UNIX hay que buscarla en su plena disposición a convertirse en la inteligencia básica de casi cualquier modelo de ordenador. Tal facultad tiene su origen en el hecho de estar escrito en un lenguaje de alto nivel que lo hace independiente de la intimidad de la máquina.



tan solo se elimina el «link» que se dirige a los mismos en el instante de la petición de información; ello se consigue por medio del comando RM.

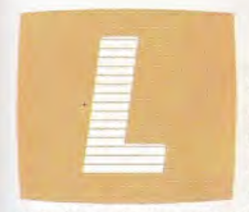
La generación de un nuevo «link» sobre un fichero ya existente se consigue haciendo uso del comando LN. Este comando, tal y como está estructurado, permite que un mismo fichero pueda ser accedido desde diferentes directorios; con lo cual, y si recordamos lo indicado anteriormente respecto a los «pathnames» y a los directorios de trabajo, evita que aquellos sean excesivamente largos al hacer que el fichero perteneciente a otro directorio se comporte de forma idéntica a otro perteneciente al actual directorio. Por medio de este comando también es posible establecer un «link» sobre un fichero del propio directorio desde este mismo, si bien, en este caso, el «link» ha de realizarse con un nombre de fichero distinto que el fichero posee. Para entender plenamente la actuación del comando LN, hay que tener presente el hecho de que éste crea punteros hacia un fichero, no genera copias del mismo y, en este caso, toda la información sobre el status del fichero, como podrían ser derechos de acceso, propietario del fichero o la información referente a la última vez que el fichero fue actualizado, permanece igual y únicamente el «pathname» es diferente, toda vez que la trayectoria por la que se accede al fichero es distinta.

Los diferentes «links» hacia un determinado fichero presentan los mismos derechos, ya que el sistema operativo UNIX es incapaz de hacer una distinción entre uno y otro. De la misma forma es posible eliminar uno de los «links» con un fichero y los demás continuarán operando de manera totalmente normal.

COMANDOS DEL EDITOR DE LINEAS «ED»

COMANDO	ACCION
e nombre	Lee el fichero nombre y lo aloja en el buffer.
n1, r nombre	Lee el fichero nombre y lo inserta en el buffer a partir de la línea n1.
n1, n2 w nombre	Escribe en el disco el fichero nombre desde n1 a n2.
± n	Se posiciona ± n líneas a partir de la línea actual.
=	Muestra número de línea en curso.
= $	Muestra número total de líneas.
n1, n2, p	Imprime desde la línea n1 a la n2.
|	Lista el contenido del buffer.
n1 a	Añade texto a partir de la línea n1.
n1 i	Inserta texto a partir de la línea n1.
n1, n2, d	Borra desde la línea n1 a la n2.
n1, n2, c	Borra las líneas indicadas e introduce texto.
/texto/	Localiza texto entre barras.
s/text1/tex2/	Cambia tex1 por tex2.
u	Anula el último cambio.
n1, n2, m n3	Mueve el bloque de líneas n1 a n2 después de n3.
n1, n2, t n3	Copia el bloque de líneas n1 a n2 después de n3.
g	Extiende el valor del comando a todo el texto.
q	Termina la sesión de edición.

UNIX (2)



os usuarios de ordenador que en algún momento desarrollan un nuevo programa o que han de introducir un volumen considerado de información en ficheros de datos, valoran en gran medida la existencia de un editor cómodo y potente.

En este aspecto, el sistema operativo UNIX dispone de un editor que puede ser considerado de los más completos; aunque al ser un editor de línea, el usuario debe enfrentarse al pequeño inconveniente de tener que direccionar cada línea individualmente. No obstante, bajo el ámbito del UNIX se han desarrollado varios editores de pantalla, en los que es posible mover libremente el cursor por todo el campo de la pantalla y realizar directamente las operaciones de edición de textos. Entre estos editores cabe destacar el «VI» desarrollado en la Universidad Californiana de Berkeley y que estudiaremos en el siguiente apartado.

Como es fácil suponer «ED» no actúa directamente sobre el fichero contenido en el disco; ello haría al editor más lento debido a los continuos accesos al disco. En la práctica, el contenido del fichero es transferido del disco a una zona de memoria, denominada «buffer», utilizada como área de trabajo durante la sesión de edición para crear o modificar el texto del fichero.

Ha de tenerse en cuenta que el contenido de esta zona de memoria es temporal: su contenido se perderá al terminar la sesión de edición si, previamente, no se ha ordenado que el contenido del buffer se escriba en un fichero.

El editor «ed» opera en dos modos: *modo comando* y *modo de inserción* de texto. En el modo comando cualquier entrada de texto se interpreta como un comando de edición, mientras que en el modo de inserción de texto la entrada de texto significa su paso al área de memoria o buffer, sumándose a su contenido.

Cuando se inicia la sesión de edición está activo el modo comando, pasándose al modo de inserción de texto al ejecutarse los comandos de insertar, cambiar y añadir texto. Para volver al modo comando basta con teclar un punto (.) seguido por un retorno de carro.

Editores de texto en UNIX

Los comandos del «ed» tienen una estructura simple y única: de cero a dos números de líneas seguidos por una letra que hace referncia al comando y, opcionalmente, los parámetros que concretan dicho comando; por ejemplo:

1,3 p

En la descripción de los números de línea están permitidos varios caracteres especiales como: el punto (.) utilizado para indicar la línea en curso, y el dólar ($), que hace referencia a la última línea del fichero.

La llamada al editor «ed» desde el caparazón o shell del UNIX se realiza con el siguiente formato:

$ ed nombre-fichero ($ es la petición o prompt del sistema, e indica que estamos en el shell).

La reacción del editor varía según exista o no el fichero especificado, respondiendo con el número de caracteres que contiene el fichero en el primer caso, o bien con el signo de interrogación seguido del nombre del fichero en el segundo. A partir de este instante el editor se encuentra en modo comando.

A continuación se relacionan los di-

El trabajo de edición se realiza sobre una zona de memoria denominada buffer. Este método acelera el proceso de edición al reducir el número de operaciones de lectura/escritura a realizar sobre el disco.

El cambio de un repertorio de caracteres por otro es tarea encomendada al comando «S» (Substitute).



La tendencia actual es facilitar al máximo la introducción de informaciones en el ordenador; tal es el cometido de los editores, procesadores de texto, periféricos para el reconocimiento de voz...

versos comandos de edición, agrupados por funciones, expresándose a continuación de la letra de comando el resto de la palabra entre paréntesis.

- *Comandos de lectura-escritura de ficheros*

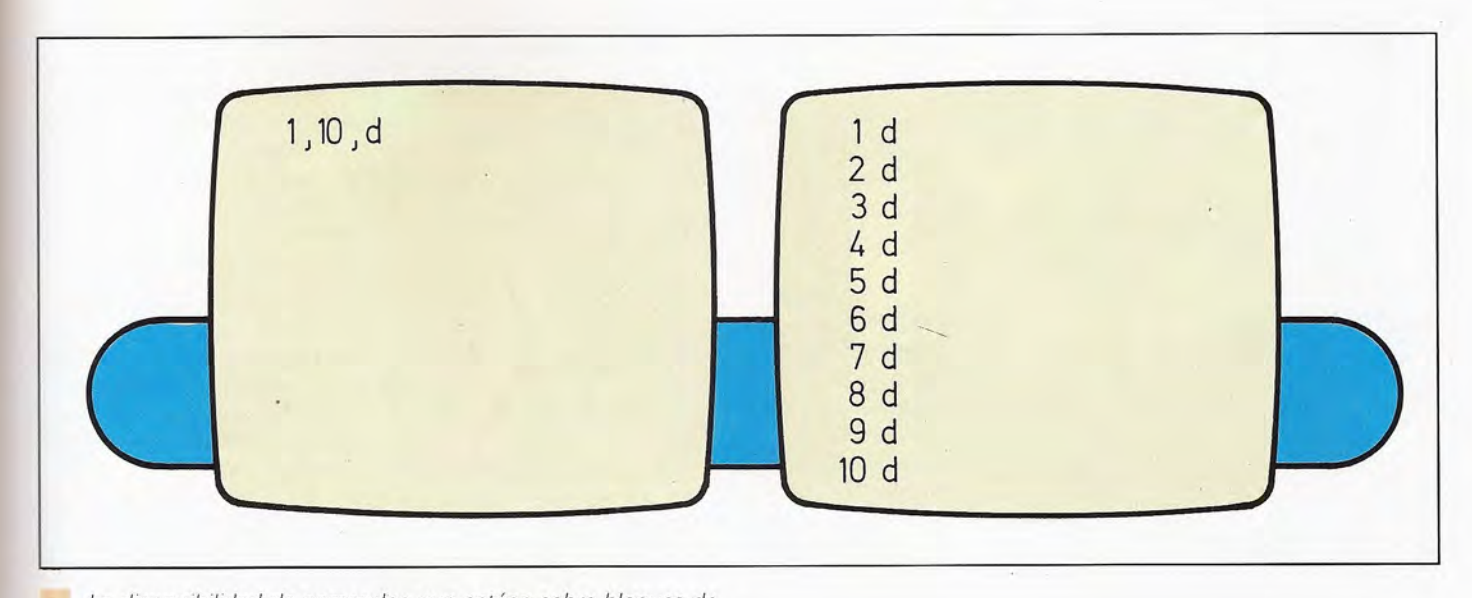
La incorporación de ficheros externos al que se está editando, o la escritura del contenido total o parcial del buffer en otros ficheros, se realiza con los siguientes comandos:

e(dit) nombre-fichero: lee un fichero y lo inserta en el buffer después de la línea n1 sin borrar el contenido.

n1,n2,w (rite) nombre fichero: escribe en el fichero especificando el contenido del buffer desde la línea n1 a n2, o el buffer completo si no se indican estos números de línea.

- *Comando de posicionamiento*

Siempre existe una línea en curso o línea activa de las múltiples contenidas en el buffer sobre la cual actúan los diversos comandos. A veces puede ser ne-



La disponibilidad de comandos que actúen sobre bloques de líneas simplifica en extremo la corrección de textos.

cesario cambiar esta línea; para ello existen los siguientes comandos:

no. línea: hace que la línea en curso sea la indicada.

+n: la nueva línea en curso es la actual menos el número n.

.=: muestra el número de la línea en curso.

$ = indica el número total de líneas del buffer.

- *Comando de listado*

La visualización del contenido del buffer se realiza con los comandos:

n1,n2p(rint): muestra las líneas comprendidas entre n1 y n2.

l(ist): lista el contenido del buffer incluyendo los caracteres ASCII y los caracteres de control existentes.

- *Comandos de inserción de texto*

La introducción de nuevo texto en el buffer es posible gracias a los comados:

n1 a(ppend): añade líneas de texto a partir de la línea n1 indicada.

n1 i(nsert): inserta líneas de texto antes de la línea n1 indicada.

Ambos comandos activan el modo de inserción de texto, siendo necesario la introducción de un punto (.) y retorno de carro (<CR>) para desactivar dicho modo y volver al modo comando.

- *Comandos de borrado de texto*

Las líneas de texto no deseadas se eliminan del buffer con el comando:

n1, n2 d(elete): borra las líneas de texto desde n1 a n2 inclusive.

- *Comandos de borrado e inserción de texto*

Dentro de «ed» cabe la posibilidad de borrar las líneas que desee el usuario y luego insertar el texto correspondiente, o bien utilizar el comando «change» que realiza estas dos funciones a la vez:

n1, n2, c(hange): borra las líneas indicadas e introduce el modo de inserción de texto.

- *Comandos de localización de caracteres*

Para encontrar una sucesión de caracteres determinados en el texto se dispone de los siguientes comandos:

/texto/: localiza la primera ocurrencia de texto en el buffer buscando a partir de la línea en curso hasta el final.

//: localiza la siguiente ocurrencia de texto buscando hacia el final del buffer.

?texto?: localiza la primera ocurrencia de texto en el buffer, buscando a partir de la línea en curso hacia el principio.

??: localiza la siguiente ocurrencia de texto buscando hacia el principio del buffer.

- *Comandos de sustitución de caracteres*

El cambio de un conjunto de caracteres por otro se lleva a cabo por medio del comando «substitute».

S(ubstituye)/texto-antiguo/texto-nuevo/: cambia el texto antiguo por el nuevo.

Para posibilitar la inserción de caracteres al principio y al final de la línea, existen dos caracteres especiales que pueden sustituir dentro del comando al texto antiguo:

^= principio de línea
$ = final de línea

si se desea anular el cambio efectuado se puede utilizar el comando «undo»

u(ndo): anula la última sustitución si se está posicionando en la línea afectada por el cambio.

- *Comandos de bloque*

La manipulación de bloques de líneas se consigue con los comandos:

10 PRINT "HOLA"
20 PRINT "SOY"
30 PRINT "SOFT"
40 PRINT "COMO ESTAS?"
50 INPUT a$

VI

10 PRINT "HOLA"
20 PRINT "SOY"
30 PRINT "SOFT"
40 PRINT "COMO ESTAS?"
50 INPUT a$

ed

La ventaja del editor de pantalla es que permite un movimiento libre del cursor a través de todas las líneas visualizadoras; ello obvia la necesidad de indicar al editor la línea sobre la que se va a actuar.

n1,n2, m(ove)n3: mueve las líneas comprendidas entre n1 y n2 situándolas tras la línea n3.

n1, n2 t(ransfer)n3: copia las líneas comprendidas entre n1 y n2 detrás de la línea n3.

- *Comandos varios*

Restan varios comandos de edición no encuadrables en ninguna de las anteriores clasificaciones; estos son:

g(lobal): hace que la acción del comando se extienda a todo el fichero.

q(uit): finaliza la sesión de edición.

!comando: ejecuta el comando del sistema sin salir del editor.

- *Caracteres especiales*

Existen dos caracteres que sirven para borrar caracteres o líneas; aunque visibles en la pantalla, estos caracteres no se incorporan al buffer:

= borra un carácter
: = borra la línea en curso.

El editor VI

El editor VI es un editor de pantalla; como tal es en algunos aspectos más potente y práctico que su compañero, el editor de línea «ed». No obstante, cabe señalar que el editor VI presenta algunos inconvenientes, tales como la imposibilidad de hacer correcciones de texto automáticamente en todo el fichero y mover a copiar bloques de líneas. Joy W. de la Universidad de California resolvió estas pegas al lograr que el editor VI se comunicase con el «ed» para realizar estas tareas hasta entonces imposibles.

La instalación del editor VI depende del tipo de pantalla a utilizar ya que por desgracia los códigos de control para el movimiento del cursor, inserción y borrado de caracteres y líneas, etc., no son iguales para todos los tipos de pantalla. En consecuencia, hay que definir las características de cada pantalla antes de poder utilizar el editor; no obstante, esto no es necesario en la mayoría de los casos al disponerse de las definiciones de las pantallas más comunes en el mercado. Esta operación sólo es necesaria cuando se instale al editor.

Una vez instalado el editor ya se está en disposición de ejecutar el abanico de comandos disponibles para VI y comprobar sus características; entre ellas destacan las de actuar en las inserciones y borrados sobre las palabras completas, así como las facilidades que ofrece para fijar el formato del texto tales como fijación de márgenes, centrado de palabras, etc.

La comunicación con el editor «ed» se realiza precediendo a cada comando «ed» con el signo dos puntos (:), con lo que se podrá actuar como si realmente la edición se hiciera desde el editor de línea.

Al igual que «ed», VI tiene acceso a los comandos shell del sistema sin necesidad de terminar la sesión de edición, anteponiendo al comando a ejecutar los signos dos puntos y exclamación (:!).

Formateando documentos

En lo que resta del capítulo hablaremos de algunas utilidades para formatear documentos. La mayoría de las posibilidades de procesamiento de texto en el sistema operativo UNIX se resuelven con las utilidades «nrof» y «trof». Desde aquí adelantamos que su uso no es absolutamente sencillo.

Antes de empezar a explicar alguno de estos programas, nos gustaría aclarar las diferencias que existen entre «formateadores de texto» y «procesadores». En el segundo las operaciones de edición y formateado se encuentran en el mismo programa y los comandos de ambas funciones se introducen desde teclado. Además, el documento ya formateado se puede visualizar en pantalla a medida que se va construyendo. En los formateadores, sin embargo, la entrada de texto y la entrada de formato son dos operaciones independientes. Primero se prepara el texto, general-

mente usando un editor de UNIX (ed, ex, vi...) y posteriormente se añaden una serie de instrucciones de formato que controlan la generación del texto final. El fichero así obtenido «pasará» a través de un programa formateador, el cual interpretará las instrucciones añadidas. El resultado de este proceso será la obtención del documento final.

La mayor ventaja de un procesador de texto es que se puede ir viendo en la pantalla la exacta representación del documento final: la división entre la edición y el formateo está oculta. Sin embargo, tiene un inconveniente. En general, un procesador de texto no puede manejar documentos de más de unas decenas de páginas. Para personas que escriben grandes manuales o libros, la elección de un procesador de texto puede resultar penosa. La realización de un cambio, como borrar una línea o insertar una nueva al comienzo de un documento, puede suponer reformatear todo el texto. Además, los procesadores de texto, en general, no te permiten numerar automáticamente los capítulos y apartados, ni tampoco te generan una tabla de contenidos de forma automática. Estas tareas tienen que hacerse manualmente y suele ser bastante engorroso, además de ser una fuente de error bastante común.

Los procesadores de texto son tremendamente útiles para escribir cartas o para manejarse con pequeños documentos, pero en cuanto se sale de ahí es conveniente el uso de formateadores.

Por otra parte, estos últimos presentan el inconveniente de que es necesario ejecutar un programa para obtener el documento final. Aunque mucha gente encuentra este proceso natural, y lo prefiere.

Actualmente, no existen procesadores de texto de uso generalizado para sistema UNIX, pero sí formateadores. El más antiguo o primitivo se denomina «roff» y los más avanzados son «nroff» y «troff». Este último se utiliza para producir salidas de fotocomposición. Algunas instalaciones soportan algunos otros más potentes, tales como «Scribe» y «TEK».

Sin embargo, los formateadores de uso más generalizado son «nroff» y «troff» y por ello serán los que expliquemos aquí.

Ideas básicas sobre «nroff»

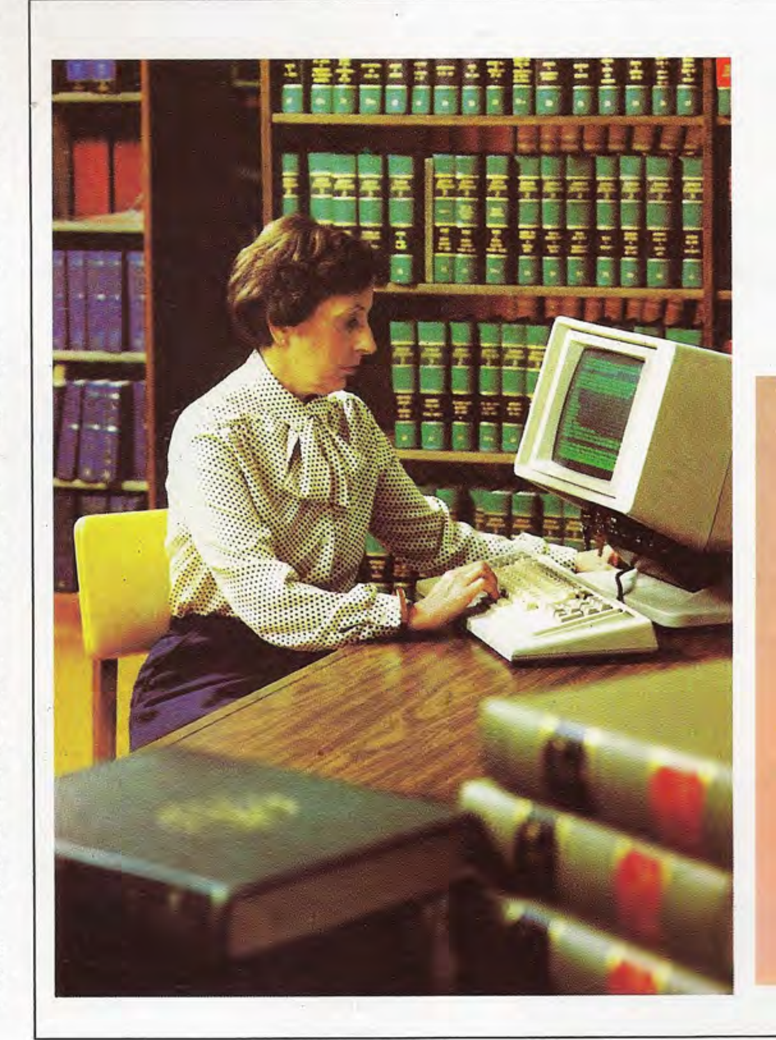
Los pasos para elaborar un documento son: primero preparar un fichero de entrada con el texto y las órdenes de formato, y después, hacer pasar a éste por el filtro «nroff».

$ nroff manuscrito ñ lpr

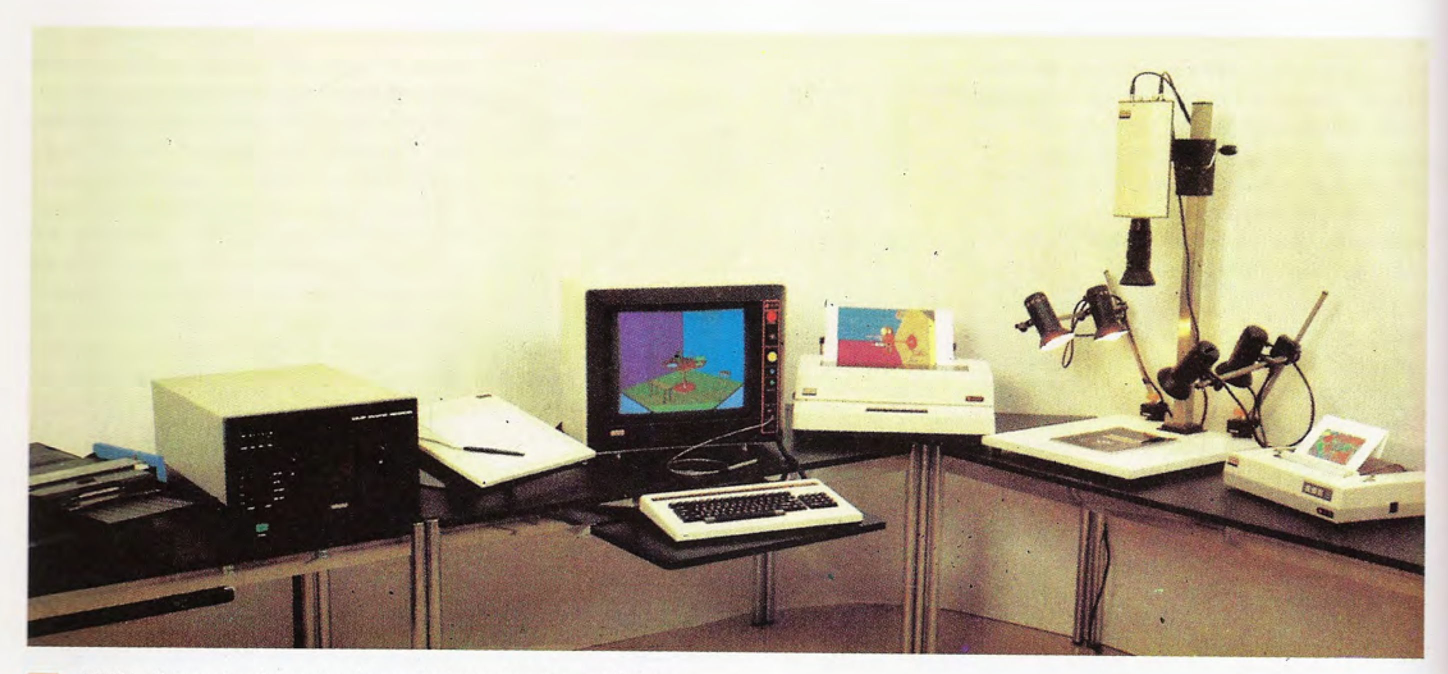
El comando «nroff» formateará el documento «manuscrito» y lo dejará listo para imprimir en la línea de la impresora. Si no se le indica a «nroff» donde ha de dirigir la salida, éste la presentará en pantalla (salida estándar). En nuestro ejemplo con la ayuda de un «pipe» («ñ») hemos redirigido el resultado a la impresora.

El fichero de entrada, en general, podrá contener o no instrucciones de formato. En caso de que no existan, «nroff» únicamente realizará operaciones de limpieza y orden. Este hará, en lo posible, todas las líneas de la misma longitud y ajustará el espaciado entre palabras de forma que estas queden pegadas a los dos márgenes laterales. No obstante, si en el texto hay líneas en blanco o se escriben blancos en una línea, éstos se dejan igual a la salida. Con ello es posible mantener la separación entre párrafos y los espacios de tabulación en la primera línea de los mismos.

Existen además otras opciones proporcionadas por «nroff» que nos permiten modificar aún más la apariencia del documento final, pero para ello ya es necesario incluir instrucciones de control en el texto. Así, por ejemplo, «nroff» nos permite seleccionar el tamaño de la



El uso de procesadores de texto es de gran utilidad cuando los documentos no sobrepasan las decenas de páginas. Los formateadores de texto, por el contrario, son de gran utilidad en estas ocasiones.

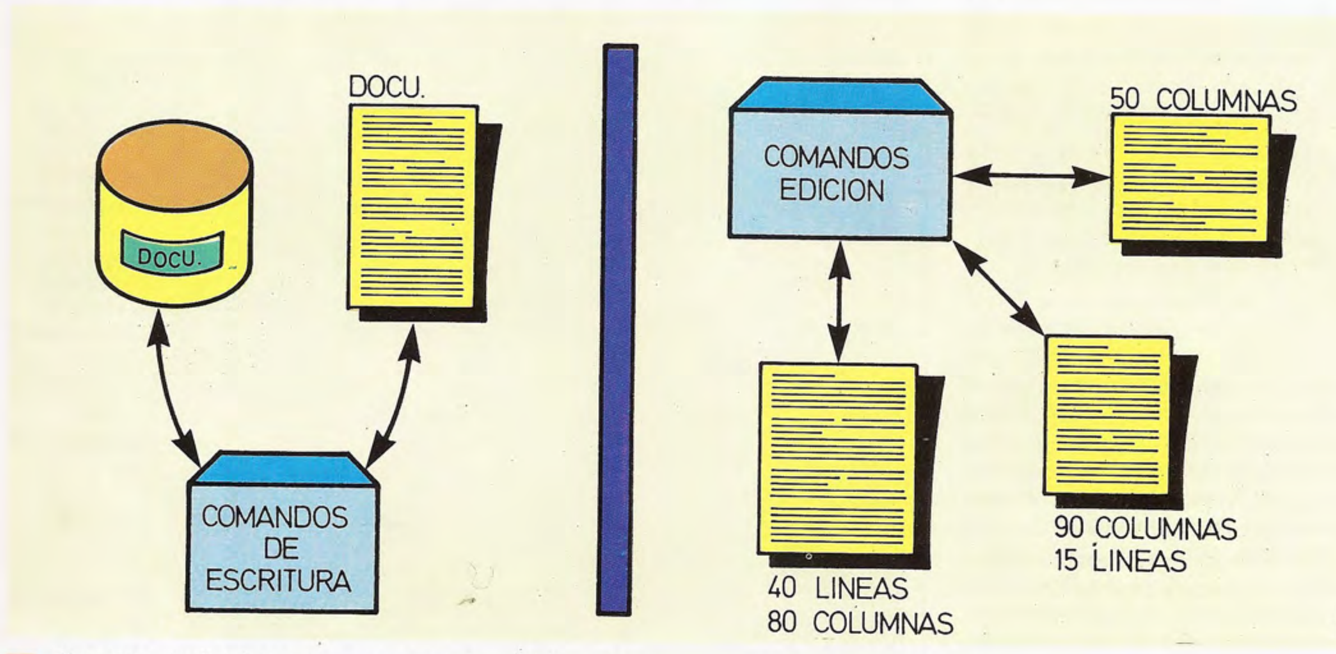


UNIX ofrece dos formateadores de texto: nroff y troff. Este último se utiliza para fotocomposición.

página, el número de líneas en la misma, el tamaño de los márgenes. También nos permite seleccionar el ajuste del texto a la derecha, izquierda, a ambos lados o simplemente que lo deje como ésta. E incluso centrar o subrayar ciertas palabras o líneas. Además, con la ayuda de «macros» prodremos disponer de notas a pie de página, paginar el documento, numerar los capítulos y secciones automáticamente y generar tablas de contenido.

Las instrucciones que nroff acepta son de muy bajo nivel. Así por ejemplo, cuando empezamos un párrafo podemos desear:

— Dejar una o más líneas de separación.



Para usar nroff primero hay que construir un fichero con el texto y añadir los comandos de formato.



Antes de obtener el documento final hay que ejecutar el comando nroff. Este ajustará el texto al formato especificado e incluirá el fichero resultante en el buffer de impresión.

— Asegurarnos de que la primera línea del mismo no se escribirá en la última de una página y el resto en una nueva.

— Iniciar la primera línea con 5 espacios.

Todas estas exigencias no podemos pedírselas a «nroff» simplemente diciéndole «empieza un nuevo párrafo»; será necesario especificarlas separadamente. Sin embargo, «nroff» nos permitirá diseñarnos nuestras propias instrucciones de alto nivel, a las que denominaremos «macros». Por ejemplo, un usuario cualquiera podrá crearse un «empieza nuevo párrafo» que contenga los tres apartados enumerados anteriormente.

La definición de una macro puede ser bastante compleja. Por ello, hay disponibles en UNIX varios conjuntos de macros ya definidos, a los que se les suele dar el nombre de «paquetes de macros». La enumeración y explicación de ellos desborda la finalidad de este capítulo. El lector interesado en estas «macros» puede consultar directamente los manuales de UNIX («nroff/troff User's Manual»).

Algunas instrucciones de «nroff»

Las instrucciones de «nroff» deben ir junto con el texto del documento en un único fichero. Estas deben aparecer al principio de la línea y no puede haber más de una por línea. Las líneas de texto tienen que ser distintas a las de control.

Una línea de control básicamente estará compuesta por una instrucción de «nroff» o la llamada a una macro, opcionalmente seguida por uno o varios argumentos separados por espacios.

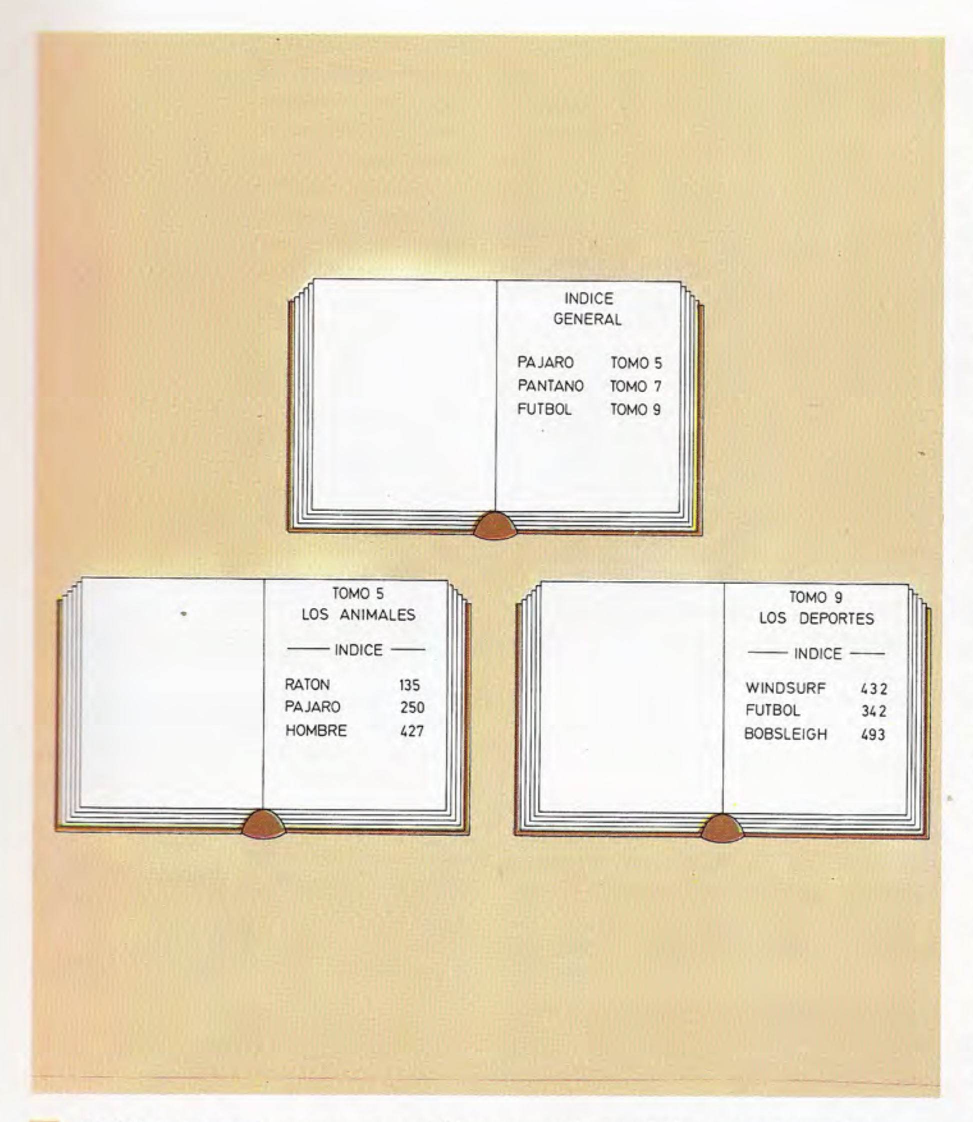
```
                    Sres. de  Abad Toribio, Maria Elena
                              Columela, 12
                              Madrid -    01


Madrid, 20 de Junio de 1.984
          Estimado amigo


                              Su hijo ha obtenido una calificación
media de     4.5
                              Es un placer para nosotros notificar-
le que, como en años anteriores, celebraremos unos cursillos es-
peciales de recuperación.
                              Con tal ayuda, confiamos en que su
hijo recuperará este verano las asignaturas que no ha superado
en Junio. No dude en consultarnos cualquier pormenor a propósito.
de tal actividad. Para todo tipo de aclaraciones, o para formali
zar la matricula, dirijase como siempre a la señorita Marian, en
la Secretaria del Centro.


                              Le saluda afectuosamente,
```

Si el fichero de entrada de nroff no contiene comandos de control, éste realizará automáticamente el ajuste a ambos lados.



Definiendo algunas macros es posible numerar los capítulos y secciones de un documento y sacar índices de forma automática.

Las instrucciones de «nroff» consistirán en un punto seguido de uno o dos caracteres. Decir también que existe una forma alternativa que empieza con comilla simple. Veamos algunas de las instrucciones con un ejemplo:

```
pl 72
po 8
bp
in + 5
ti -3
'ul
IP "first:" 14
```

La última línea es una llamada a una macro, y el resto son instrucciones básicas de nroff. Como se observa en el ejemplo, una macro tiene la misma apariencia que una instrucción regular, excepto quizás en el uso de letras mayúsculas. Aunque no es obligado, se suelen definir las macros con estas letras, para distinguirlas de las expresiones regulares que siempre tienen que ir en minúscula. Conviene decir también, por si el lector se encuentra alguna vez con casos parecidos, que el uso de un espacio para separar el primer argumento no es obligado, aunque conviene para una mejor comprensión.

La forma que los argumentos pueden tomar depende de la instrucción o macro, pero generalmente suelen ser numéricos. Hay dos posibles tipos de argumentos numéricos: un número sin signo —lo que significa que se toma como valor absoluto— o un número precedido de un signo —lo que significa una variación relativa respecto al valor actual del parámetro en cuestión—.

Veamos ahora algunas de las instrucciones más usadas

— Se puede modificar la distancia entre el margen derecho del papel y el margen derecho del texto con '. po Pt ' (Pt representa el número de caracteres a separar).

— Se puede modificar la longitud en caracteres de la línea con '.pn+ll Pt' y la longitud en líneas de una página con 'pl Pt'.

— nroff no proporciona ruptura y numeración de páginas automáticamente, sino que es necesario definirse unas macros para ello. Con '.pn+1' podemos forzar la numeración y con '.bp' la ruptura.

— Si deseamos asegurar que un párrafo no se rompa al final de una página sin al menos tener en la misma un número mínimo de líneas, usaremos '.ne Pt'.

— En condiciones normales «nroff» ajusta las líneas a ambos extremos. Si por alguna razón especial deseamos que este ajuste no se realice, usar '.na'. Si queremos que se realice un ajuste solo a la izquierda, '.ad l'. Análogamente para la ajuste a la derecha, '.adr'. Si deseamos que la línea quede centrada, '.adc'. Para volver al ajuste normal usar '.adb' o '.adn'.

— Para ajustar el espaciado entre líneas usar '.Ps Pt'. Y en caso de que deseemos introducir líneas en blanco usar '.sp Pt'.

— Para resaltar títulos muchas veces se recurre a subrayarlos o a centrarlos en la línea. Ya hemos visto que con '.adc' se pueden cambiar el ajuste de las líneas de texto, pero cuando únicamente se quieren centrar unas pocas, se suele utilizar '.ce Pt'. Para subrayar una línea se usa '.ul Pt'. En ambos casos Pt representa el número de líneas a partir de la actual en las que ha de realizarse la operación.

UNIX (3)

La herramienta Shell



no de los puntos básicos de un sistema operativo se concreta en su comunicación con el usuario del ordenador. La interacción debe ser sencilla y cómoda para el usuario, con las consiguientes ventajas a la hora de desarrollar programas o, simplemente, de recabar información de las entrañas del equipo. Algunos elementos del sistema operativo que dan paso a esta eficaz interrelación ordenador/usuario son:

— Comandos con nombres claros y constituidos por una sola palabra.

— Ausencia o número limitado de parámetros en el comando.

— Posibilidad de emplear abreviaturas en los nombres de los comandos.

— Mensajes de error claros y cubriendo el mayor número posible de errores distintos.

Un sistema operativo con estas características podría ser definido como «amigable» para el usuario, toda vez que le proporciona una gran ayuda en su trabajo, descargándole de la necesidad de memorizar largas secuencias de comandos y parámetros.

En el extremo inferior de esta escala de bondad se encuentran sistemas operativos como el CP/M y el MS-DOS, los cuales presentan un repertorio de comandos más o menos farragoso y con múltiples parámetros. En el extremo opuesto aparece el sistema operativo de los ordenadores LISA y MACINTOSH de Apple, en los cuales toda la gestión se realiza a través de menús de tipo persiana y símbolos gráficos (iconos) por pantalla; no se hace empleo de comandos tecleados, en lo que cabe catalogar como un ejemplo máximo de facilidad en la interface ordenador/usuario.

El sistema operativo UNIX se encuentra a mitad de camino entre ambos extremos. Por su propia concepción, el UNIX dispone de una herramienta denominada Shella través de la cual se gestionan los diferentes comandos que se dan al ordenador. Con esta utilidad el sistema puede acondicionarse a las necesidades de los distintos usuarios, estableciendo, en caso de necesidad, una estructura en forma de menús de manera que a través de éstos sea posible el acceso a distintos comandos o programas de aplicación. Por esta causa, el UNIX es un sistema operativo cómodo para el usuario; sin llegar a la facilidad de manejo propia del sistema operativo que gobierna a los ordenadores LISA y MACINTOSH presenta unas características sumamente prácticas para el usuario. Un extremo asociado a esta constatación, también importante, lo constituye el hecho de que las diferentes interfaces de los usuarios con el ordenador pueden ser distintas entre sí, con lo cual el sistema operativo puede parecer distinto a cada usuario, no estando obligado por tanto a una interface estándar que podría no resultar totalmente adecuada a sus necesidades.

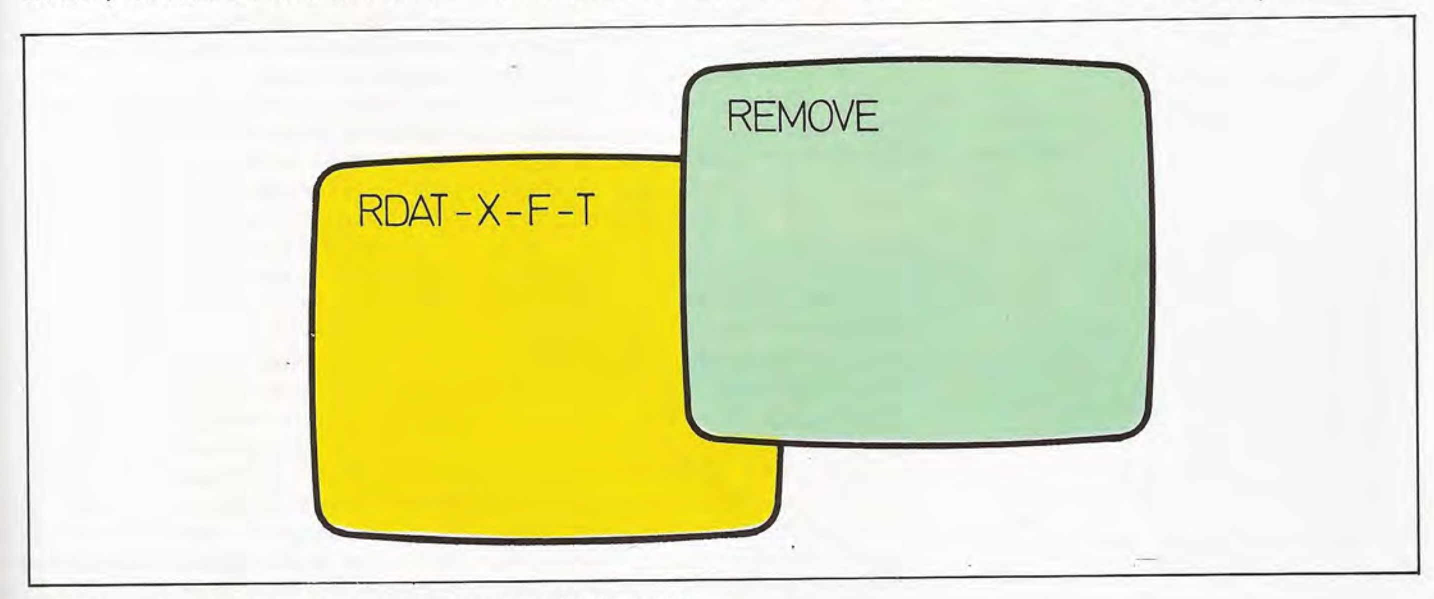
En la actualidad existen dos tipos de Shell que cabe considerar como los más comunes: el Shell Bourne, correspondiente al sistema operativo UNIX estándar, y el Shell C, correspondiente al UNIX desarrollado en Berkeley. Esta utilidad, el Shell, no es sólo una herramienta para la gestión de los distintos comandos; además, es un lenguaje de programación de alto nivel, a través del cual se pueden gestionar las llamadas a distintos programas empleando e implementando una estructura «pipeline» en la cual la salida de un programa es utilizada como entrada de otro.

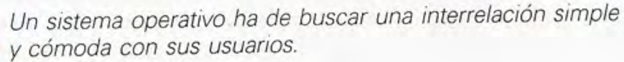
La línea de comandos

La ejecución de cualquier programa se lleva a cabo indicando al sistema operativo un comando. La línea que contiene el comando, incluyendo los distintos argumentos, se denomina línea de comandos. Su sintaxis es la que determina la ortografía, puntuación y distribución de los distintos elementos que intervienen en un comando.

La estructura general de una línea de comando toma el siguiente aspecto:

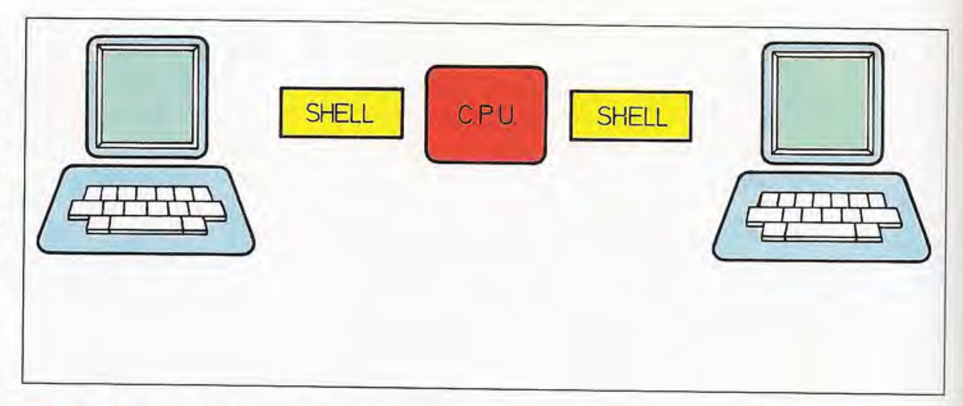
comando (arg 1) (arg 2)... (arg w)



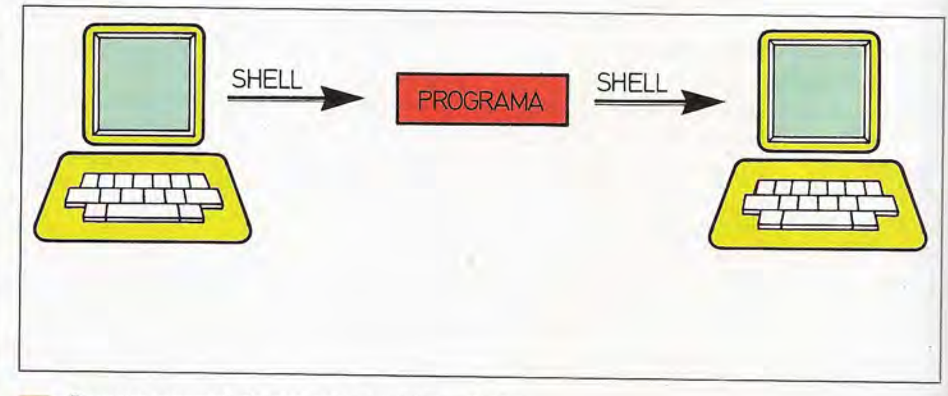

Un sistema operativo ha de buscar una interrelación simple y cómoda con sus usuarios.

Los diferentes argumentos de un comando no son estrictamente necesarios; su necesidad o no depende exclusivamente del comando en cuestión y de la función que éste lleve a cabo. El argumento como tal puede estar compuesto por un texto, número, nombre de un fichero o, en general, por cualquier dato que indique al comando sobre qué tiene que actuar. Como ejemplo más simple podríamos indicar que en el caso de un comando de edición, un argumento podría ser el nombre del fichero sobre el cual va a trabajar. Dentro de los argumentos están los denominados opciones, los cuales modifican los efectos del comando, no existiendo limitación en el número de opciones a incluir en el conjunto de un comando. Su estructura va siempre precedida por un guión en el caso de los programas de utilidad, pero no en el Shell en sí. El tratamiento de cualquier comando se realiza a través de un buffer intermedio en el cual se van almacenando los distintos caracteres que constituyen la línea de comandos. Los caracteres van siendo analizados individualmente, en espera de encontrar algún carácter de control, como pueda ser una indicación de eliminar caracteres o incluso una línea completa de comandos. Caso de no ser así, éstos son almacenados en el buffer; una pulsación de la tecla RETURN hará que el contenido del buffer, coincidente con el conjunto de la línea de comandos, sea enviado al Shell para que éste lleve a cabo el tratamiento del conjunto.

El Shell lleva a cabo un proceso hasta cierto punto inverso al anterior. Mientras que inicialmente se efectuaba una carga en el buffer, ahora se realiza una descomposición de la línea en sus distintos componentes. El comando es identificado buscando el grupo de caracteres localizado hasta el primer espacio. Caso de que este nombre no coincida con ninguno de los conocidos, se emitirá un mensaje de error. Una particularidad del Shell es el hecho de que no tiene por sí mismo capacidad de discurrir si una opción particular o cualquier otro argumento es válido para un programa determinado. Los mensajes de error correspondientes a estas situaciones han de ser considerados por el propio programa, dándose el caso de que muchos programas de utilidad UNIX ignoran las opciones erróneas.



El shell es el elemento de conexión entre los deseos del usuario y la máquina.



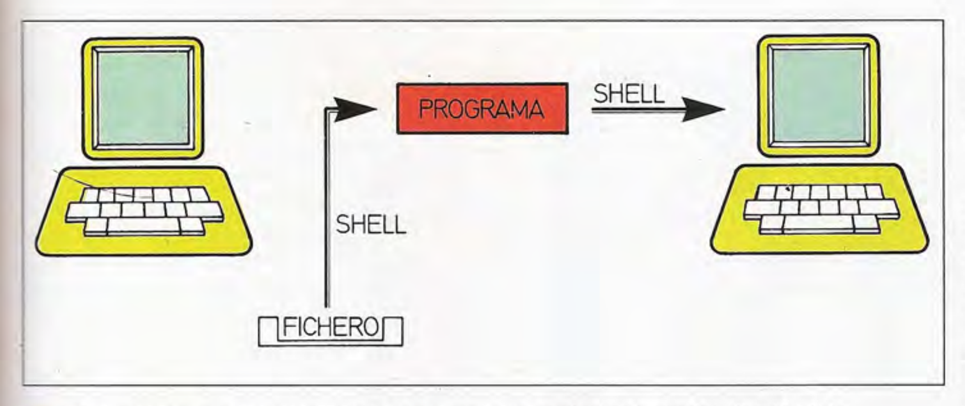
Proceso normal de entrada y salida estándar.

La ejecución del comando se inicia una vez que el Shell ha encontrado un programa con el mismo nombre. En este momento pasa las opciones y argumentos al programa y éste comienza su ejecución. Mientras ésta se produce, el Shell entra en un estado «durmiente» hasta que recibe una señal indicando que el programa ha finalizado su ejecución, instante en el que pasa de nuevo a un estado activo en el cual queda en disposición de gestionar nuevos comandos.

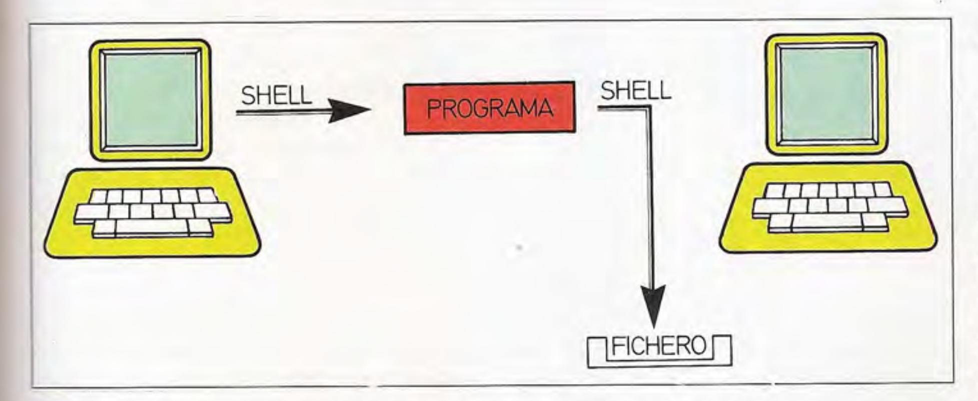
Entrada y salida estándar

Cualquier comando del sistema operativo UNIX lleva a cabo la gestión de los distintos elementos de información contenidos en la memoria del ordenador. En algunos casos la respuesta a un comando es una lista de información, la cual ha de llegar al usuario de una u otra forma. El lugar al que se envía la información es lo que se denomina salida estándar; normalmente el programa nunca sabrá donde se envía esta información, si es enviada a un terminal o a un fichero. De igual manera encontramos que todo comando ha de recibir sus órdenes de algún sitio, bien sea de un fichero o a través de un terminal. Este conjunto es lo que se conoce por entrada estándar.

Hemos visto la estructura de los directorios y ficheros en el sistema operativo UNIX. Sin embargo, ha de señalarse la presencia de un tipo de fichero denominado «fichero de periférico». Este tipo de fichero es residente en la estructura del sistema operativo UNIX y representa a un dispositivo periférico como puede ser un disco, un terminal o una



Proceso de redireccionamiento de la entrada estándar (teclado) dirigiéndola hacia un fichero residente en memoria externa.



Redireccionamiento de la salida estándar (pantalla) hacia un fichero.

impresora. Este fichero puede ser empleado como entrada y salida estándar de los distintos comandos y programas. Así, en el caso de un terminal, cuando un usuario se conecta al sistema, el Shell dirige la salida estándar al fichero de periférico que representa a su terminal, de forma que todas las informaciones de salida aparecen en el terminal. Igualmente, el Shell dirige la entrada estándar para que provenga del mismo fichero, de forma que cualquier cosa tecleada en el terminal es un «input» del programa.

Este tipo de operación sería muy rígida si no fuera porque el Shell puede redirigir la entrada y salida de cualquier programa. Esta redirección se realiza asociando la salida y entrada estándar de cualquier programa con un fichero distinto del fichero de periférico correspondiente al terminal que se esté empleando. La redirección de la salida se realiza por medio del símbolo de redirección de salida (>), el cual va seguido por el nombre del fichero sobre el que va a producirse la salida.

La forma global de esta redirección sería la siguiente:

Nombre de programa (argumentos)> nombre de fichero salida

La redirección de la salida ha de realizarse con cuidado, toda vez que el Shell escribe sobre el fichero al que se redirige borrando su contenido, el cual se pierde. Para obviar esta posibilidad, o bien para llevar un control de las diferentes salidas sucesivas, es posible redirigir la salida de manera que la misma se añada al final del fichero de salida estándar correspondiente, no alterando la información previamente grabada. El símbolo que se coloca en lugar del empleado para la redirección de la salida es >, manteniéndose un comando con una estructura totalmente similar a la señalada anteriormente.

La entrada estándar también es posible modificarla, de manera que un programa determinado acepte la información de entrada de un fichero en lugar de obtenerla del terminal. El proceso a seguir es similar al señalado anteriormente, con la única diferencia de emplear ahora el símbolo de redirección de la entrada (<).

El formato del comando queda ahora como sigue:

Nombre del programa (argumentos) < nombre del fichero de entrada.

Estructura del pipe

El sistema operativo UNIX aprovechando la facilidad que el Shell proporciona al usuario para la redirección de la información, permite el crear una estructura de pipe o pipe-line en la cual es posible el empleo de la salida estándar de un programa como entrada estándar de otro; de esta forma la información generada por un proceso puede ser empleada por otro a continuación. El formato que toma esta estructura es el que sigue:

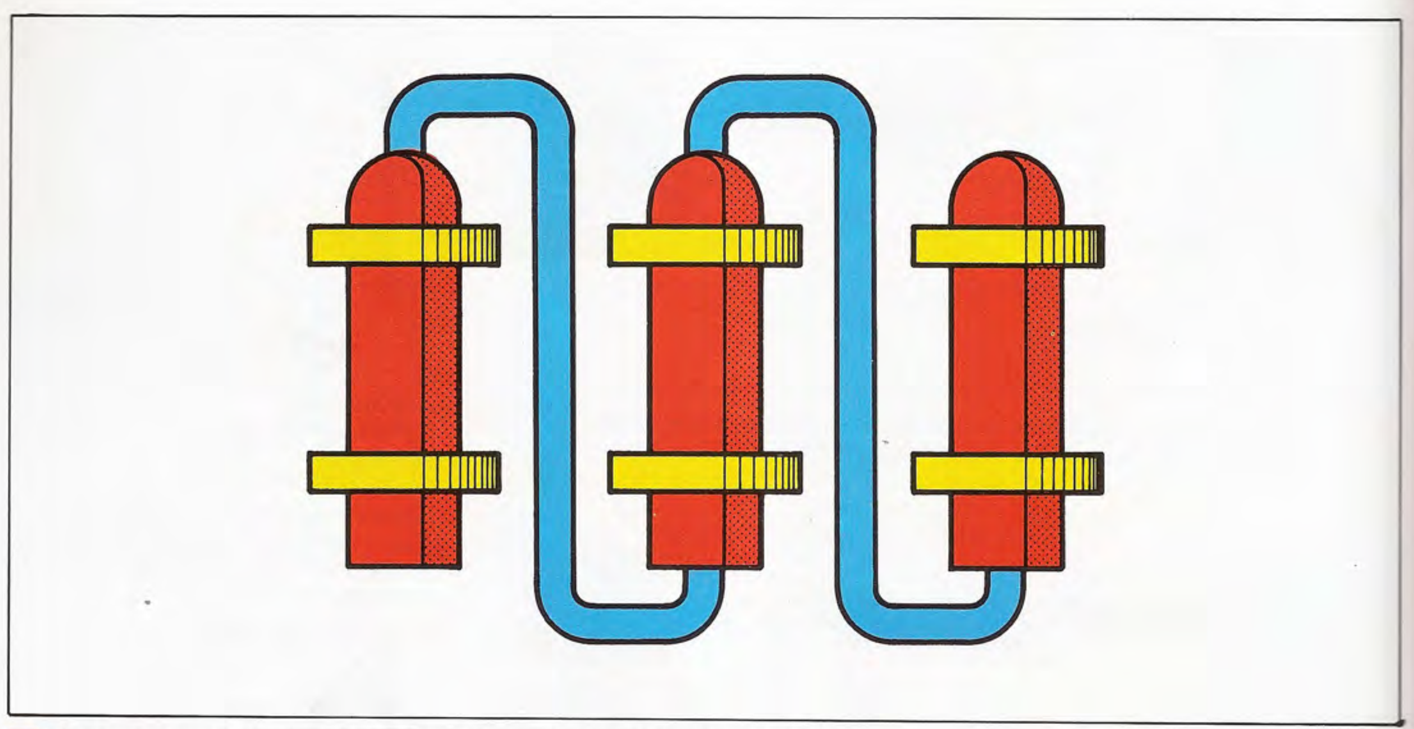
Nombre de programa 1 (argumentos) | nombre del programa 2 (argumentos)

Como se observa, la separación de los nombres se lleva a cabo por medio de una línea vertical.

Una posibilidad que da la estructura «pipe» es la de emplear lo que se denominan filtros. Un filtro es un programa que utiliza un vector de datos de entrada para producir un vector de datos de salida. Una línea de comandos que emplee un filtro utilizará la salida estándar de un programa como entrada del filtro y, posteriormente, la salida del filtro como entrada de otro programa.

Ficheros ejecutables

Como se indicó al comienzo del capítulo, el Shell es simultáneamente un in-



En la estructura «pipeline», la salida de un proceso constituye la entrada de otro.

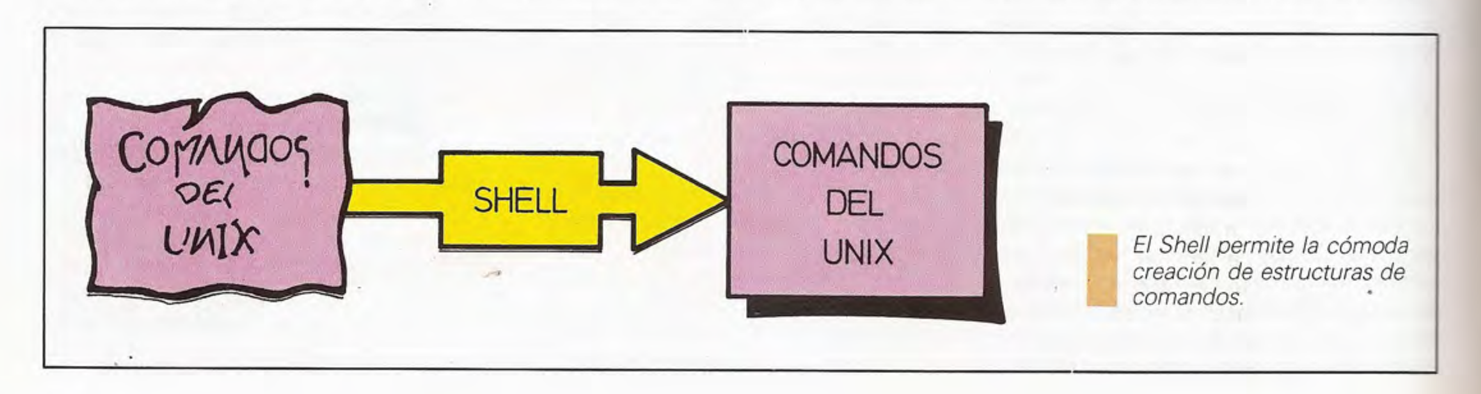
térprete de comandos y un lenguaje de programación de alto nivel. Como intérprete de comandos procesa los que le son introducidos en respuesta a la señal que evidencia que está dispuesto a aceptarlos. Cuando actúa como lenguaje de programación procesa grupos de comandos almacenados en ficheros denominados «Shell scripts». En los próximos párrafos se tratará de todo lo referente a este tipo de operación del Shell, empleando para ello los esquemas referidos al Bourne Shell que, en gran medida, son similares los del C Shell.

Una particularidad de los ficheros «Shell scripts» es permitir que diferentes líneas de comandos puedan ser agrupadas de forma que un único comando pueda ejecutarlas. Por medio de este tipo de estructuras es posible que un usuario del sistema operativo UNIX ejecute tareas complejas, y en muchas ocasiones largas y tediosas, de una forma simple.

A través de este tipo de comandos es posible copiar varios ficheros, cambiar su nombre y borrar otros, cosa común en un proceso de actualización y reajuste de los contenidos de la memoria de un ordenador por parte del administrador del sistema, por medio de un único comando. Igualmente, es posible redirigir la entrada y salida estándar dentro de un Shell Script, de manera que puedan combinarse programas de utilidad UNIX en forma adecuada a las necesidades del usuario.

Se conoce como fichero ejecutable a aquel fichero que cualquier usuario es capaz de ejecutar, o mejor dicho, que cualquier usuario tiene permiso para ejecutar. Normalmente, contiene un programa compilado o un Shell Script.

Para tener acceso a estos ficheros es necesario, por una parte, que a través del editor sea posible acceder a los mismos y por otro lado, es preciso tener derechos de ejecución sobre ellos. Caso de



El Shell permite la cómoda creación de estructuras de comandos.

no ser así, por medio del comando *chmod* es posible cambiar los privilegios de acceso asociados al fichero y hacer que un usuario tenga privilegio de acceso sobre el mismo. Una vez que el usuario ha adquirido estos derechos, tecleando el nombre del fichero directamente, éste será reconocido como un comando y ejecutado de inmediato. Si son varios los usuarios del comando, será preciso establecer estas prioridades y derechos de acceso para cada uno de ellos. Estos ficheros, además de estructuras de control, admiten variables como las señaladas seguidamente.



A través de un file-script es posible ejecutar muchos comandos a partir de una sola orden.

Variables

El Shell es capaz de aceptar variables de cadena (variables que son capaces de tomar el valor de una cadena de caracteres) de forma que puedan ser almacenados números y texto. De estas variables existen tres tipos distintos:

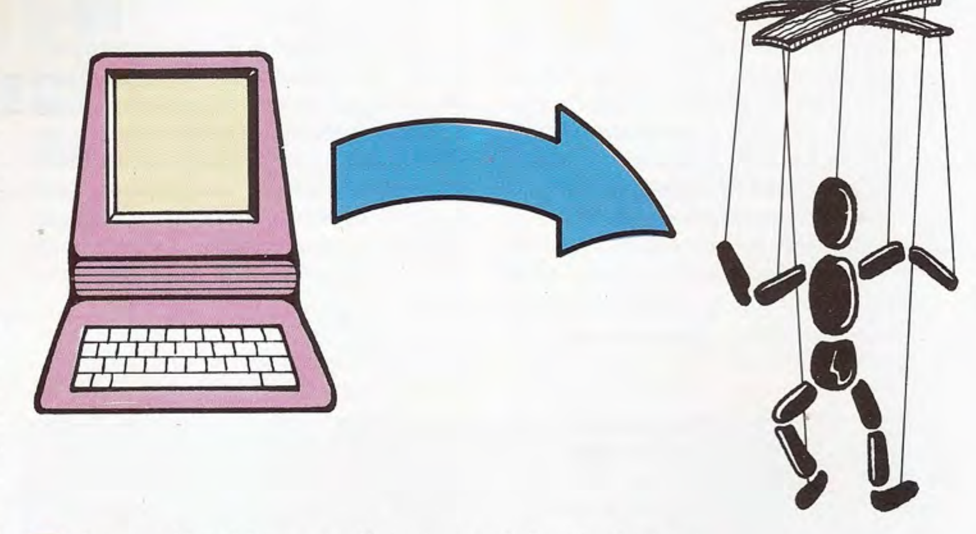
— Variables de usuario
— Variables de Shell
— Variables de Shell de solo lectura

Las variables de usuario pueden ser declaradas, inicializadas, leídas o modificadas desde la línea de comandos o bien desde un Shell Script. El Shell se ocupa de declarar e inicializar las variables de Shell, pero el usuario tiene la facultad de leerlas y modificarlas; en el caso de las variables de Shell de sólo lectura, él mismo las declara e inicializa pero no permite el cambio, aunque sí la lectura por parte del usuario.

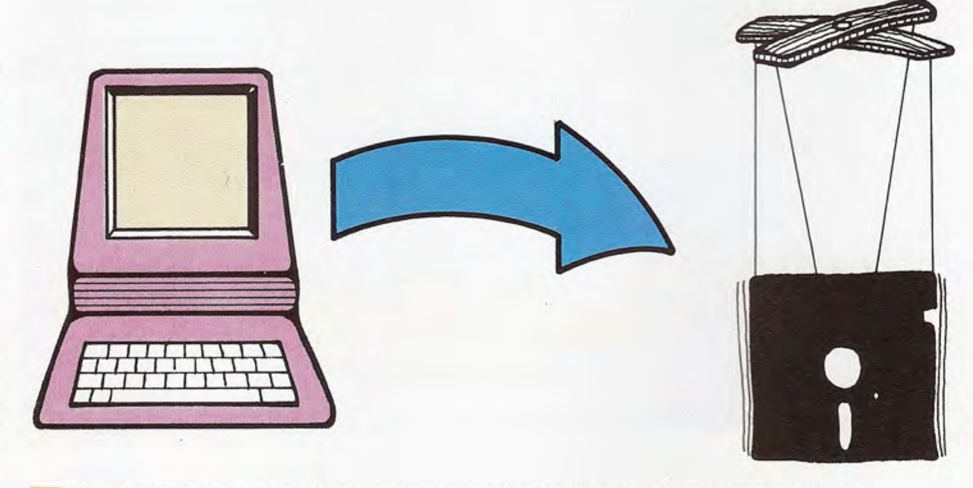
• *Variables de usuario*
Cualquier secuencia de caracteres distintos de espacios en blanco puede ser declarado como nombre de una variable, inicializándola seguidamente con un valor determinado. Así, por ejemplo:

nombre = Juan

En este caso se ha declarado una variable bajo la denominación nombre, variable que ha sido inicializada con el valor Juan. Un dato a resaltar es el hecho de que el signo igual en el comando que acabamos de señalar no puede ir precedido ni seguido por un espacio en blanco. Caso de desear que en la asignación



Las variables del usuario sólo son controladas por el mismo.



Las variables de Shell quedan bajo el control exclusivo del sistema operativo.

de valores a la variable aparezcan espacios en blanco, será necesario encerrar a la misma entre comillas.

nombre="Juan Martín"

Si el usuario deseara conocer el contenido de una variable tendría que hacer uso del comando *echo*. Sin embargo, el uso del comando *echo* directamente no daría resultado. Por ejemplo:

echo
nombre
echo $ nombre
Juan

De la primera línea de comando resulta que *echo* aplicado directamente produce tan sólo la visualización del nombre de la variable. Si bien, cuando el nombre de la variable se expresa precedido por un signo $ es cuando se presenta la cadena nombre con el valor Juan. La función del signo $ es la de comunicar al Shell que la palabra que sigue al signo $ es el nombre de una variable. En estas condiciones el Shell toma el nombre de la variable, lo sustituye por su valor, y pasa éste para ser procesado.

El usuario tiene la posiblidad de inhibir este proceso simplemente colocando entre comillas el nombre de la variable precedido por el signo $, con lo cual no se producirá la asignación de valor a la variable. Así, por ejemplo:

echo "$ nombre"
nombre

- *Variables de Shell de sólo lectura*

Las variables de Shell de esta índole son variables que se declaran como de sólo lectura y a las que ha de asignarse un valor antes de ser declaradas como de este tipo. Una vez producida la declaración, este valor ya no puede ser cambiado; de intentarlo, el Shell producirá un mensaje de error. La asignación de valores a este tipo de variables es llevada a cabo directamente por el Shell, y la declaración de su carácter se realiza de la forma indicada en el ejemplo: especificando el nombre de dicha variable y anteponiendo la sentencia *read only.*

read only nombre



Por medio de los ficheros «shell-script», el usuario puede ejecutar secuencias de múltiples comandos activando una sola orden.

- *Variables de Shell*

El Shell declara e inicializa una serie de variables a través de las cuales es posible definir la forma de *prompt* del sistema que el usuario va a recibir, la trayectoria de la búsqueda que el Shell sigue cuando recibe un comando, o el directorio base del usuario. Estas variables no son fijas sino que son actualizables, bien sea desde la línea de comando o desde el fichero profile situado en el directorio base del usuario. A continuación vamos a señalar cuáles son estas variables, así como su funcionamiento básico.

- *Variable Home*

Esta variable se emplea para almacenar en la misma el nombre del directorio de trabajo en el cual el usuario va a operar. Por defecto, el directorio base es el directorio de trabajo cuando se conecta al usuario al sistema por vez primera, este valor queda almacenado en la variable HOME. Si su valor se modifica especificando en ella un nuevo directorio, cuando se ejecute el comando CD para cambiar de directorio —si no se ha indicado ningún parámetro en el mismo— se producirá el cambio de directorio actual al señalado en la variable HOME. Así, por ejemplo:

echo $ HOME
/usr/ JUAN
cd
pwd
/usr/ JUAN

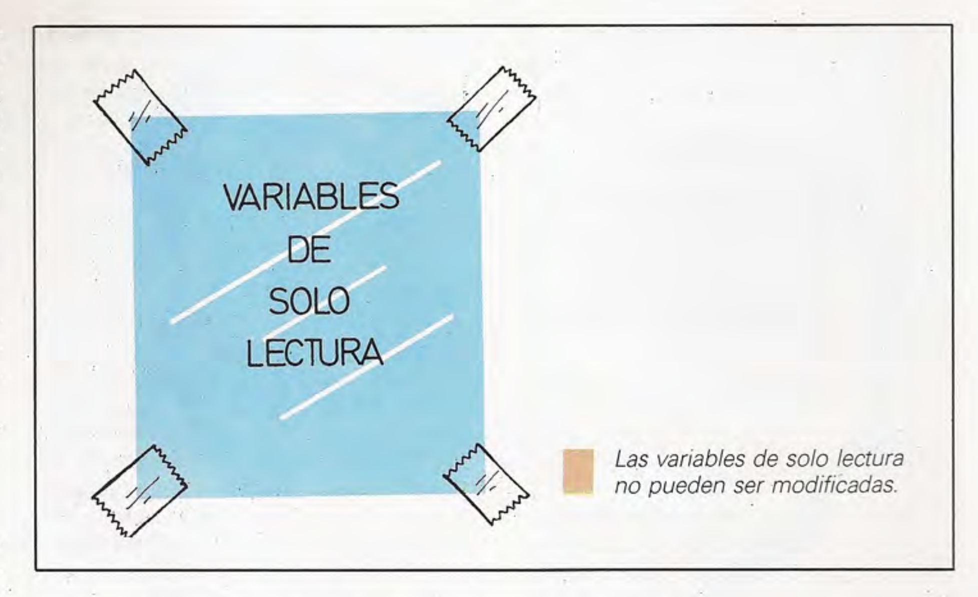
- *Variable PATH*

Cuando al Shell se le da un comando, este ejecuta una búsqueda a través de la estructura de ficheros para localizar el que contiene el programa que se desea ejecutar; en su búsqueda controla diferentes directorios hasta encontrar el programa, comenzando por el directorio de trabajo y continuando con los directorios / bin y /usr / bin. En el caso de que la búsqueda sea infructuosa en estos directorios, el Shell informaría al usuario de que no ha podido localizar el programa para que éste pueda tomar la acción correctiva pertinente.

La función del PATH es almacenar los directorios especificando de esta forma el orden en el que han de ser rastreados; dicho orden es el correspondiente a los nombres de los diferentes directorios contenidos en el PATH, leídos de izquierda a derecha.

$ PATH = :/ usr/Juan: /usr/bin :/bin

Si a cada usuario se le asigna un valor de PATH diferente, cada uno ejecutará un programa distinto aunque introduzca la misma orden.



Las variables de solo lectura no pueden ser modificadas.

- *Variables PS1 y PS2*

Cuando el Shell está en disposición de recibir un comando presenta un signo en la pantalla que revela este estado. Este signo no es fijo, sino que puede ser variado por el usuario y es almacenado en forma de cadena en la variable PS1. Cada vez que el valor de esta variable se modifique, así lo hará el prompt del sistema. Caso de que un comando no esté completo en una línea y haya necesidad de continuar en la siguiente, aparecerá en ésta el prompt o indicador secundario. El prompt secundario se almacena en la variable PS2 y al igual que sucede con la variable PS1, cada vez que es modificada el prompt secundario cambia en consecuencia.

- *Variable IFS*

Normalmente, cuando se escribe un comando o sus argumentos se emplean espacios en blanco para separar los mismos. Sin embargo, por medio de la variable IFS es posible hacer que el símbolo almacenado en la misma se con-

Visión artificial

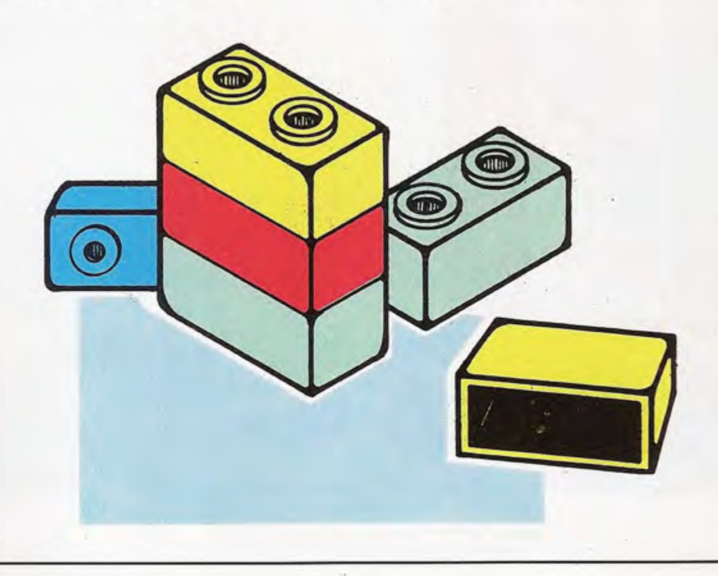
La vida real ha presentado a la tecnología multitud de retos en el deseo de esta última de repetir o imitar fenómenos de la Naturaleza. Uno de los fenómenos naturales básicos, y que sin embargo presenta a la tecnología un problema de enorme magnitud, es el fenómeno de la visión.

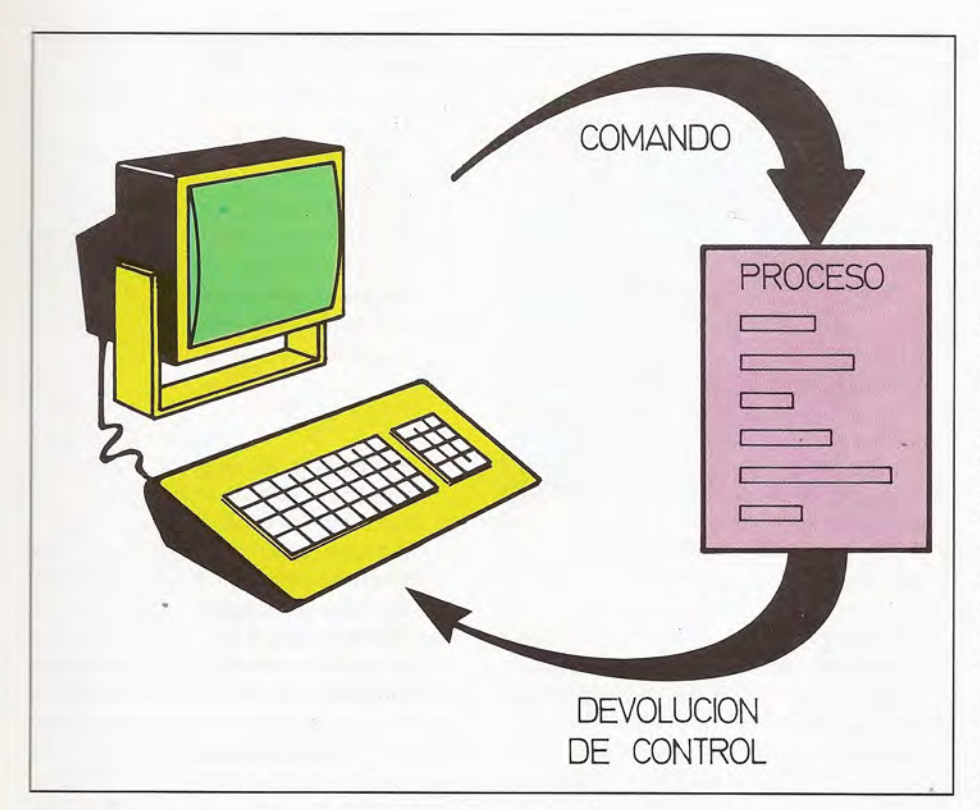
La respuesta tecnológica a este fenómeno ha dado muchos resultados diferentes, pero hasta ahora los sistemas puestos en operación y capaces de «ver» son sistemas que trabajan en contextos muy específicos y limitados, normalmente en la inspección de circuitos integrados o en la manipulación de piezas dentro de una cadena de montaje. Este tipo de sistemas operan normalmente sobre imágenes binarias (en blanco y negro), no pudiendo aceptar la escala de grises que produciría el empleo de una cámara normal de televisión.

El problema que en la actualidad se intenta resolver y que es algo fundamental en la vida real, es el de la selección de objetos en un recipiente con distintos elementos. Los sistemas de manipulación anteriormente mencionados trabajan sobre objetos situados separadamente sobre una placa horizontal, mientras que ahora se trata de escoger entre las distintas piezas del recipiente la más adecuada a la operación que se lleva en curso. Los ordenadores que controlan este proceso han de ser sumamente rápidos, para poder dar respuesta al usuario en un intervalo de tiempo razonable.

En la actualidad, partiendo de estudios médicos sobre el fenómeno de la visión humana, se están desarrollando ordenadores para este tipo de actividades. Internamente, están organizados para gestionar los procesos lógicos asociados con la visión según una arquitectura de procesadores en paralelo, en la que distintos procesadores tratan simultáneamente diferentes parcelas de la información.

Pese a que ya en los primeros años 60 Larry Roberts, en el MIT, puso los cimientos al problema de la comprensión de imágenes estableciendo una serie de métodos de análisis, y pese al gran avance tecnológico producido desde entonces, la idea de que una máquina pueda comprender imágenes de la vida real como una persona parece, aún lejana.





Cada vez que se lanza la ejecución de un comando se inicia un proceso.

vierta en un separador de campos. Así, si IFS se hace igual a «:», un conjunto como pudiera ser a:b:c:d se convierte en cuatro datos distintos (a, b, c, d) en lugar de un conjunto de caracteres únicos (a:b:c:d)

Procesos

Los procesos son los medios a través de los cuales el sistema operativo UNIX ejecuta los comandos iniciándose un proceso cada vez que se da una orden, aunque esto también puede suceder a instancias del sistema operativo.

La estructura de los procesos en el sistema operativo UNIX es jerárquica, al igual que las estructuras de ficheros. Existe una raíz que está constituida por un proceso de «login» para terminal, que se inicializa cuando el usuario se conecta al terminal. Cuando se da un comando al sistema operativo, éste genera un proceso hijo para ejecutar el mismo, y mientras tanto, el proceso que podríamos denominar padre permanece en estado durmiente o inactivo. Cuando el proceso hijo finaliza, el proceso padre despierta y toma el control, indicando al usuario que está dispuesto para recibir nuevas instrucciones. En el caso de que un proceso deba actuar sin intervención del usuario, se genera el proceso hijo, pero sin pasar el proceso padre a estado durmiente y manteniéndose activo. Para lograr un control adecuado, el sistema operativo UNIX asigna a cada proceso un número por medio del cual los procesos se identifican en tanto en cuanto estén operativos.

Traducción por ordenador

En la mentalidad popular el ordenador es una máquina orientada esencialmente a la ejecución de cálculos matemáticos y, como tal, cabría clasificarlo como una máquina matemática. Sin embargo, la realidad es que se trata de una máquina lingüística, ya que su misión básica es la de interpretar una serie de símbolos lingüísticos que constituyen los programas que dan vida a la máquina; símbolos lingüísticos llenos por tanto de significado. Un objetivo envidiable sería poder emplear el propio lenguaje hablado convencional para poder llevar a cabo estas tareas. Ello se ha intentado repetidamente; un botón de muestra se encuentra en las actuales máquinas traductoras, las cuales permiten el paso de un lenguaje natural a otro. Dichas máquinas han sido un subproducto de un problema informático muy superior. La posibilidad de la traducción por ordenador surgió en el año 1949, cuando los pocos ordenadores existentes en el mundo se encontraban en instalaciones militares. En esta fecha, el matemático Warren Weaver estableció que las técnicas desarrolladas para la descripción de códigos podían ser aplicadas a la traducción mecánica. El proceso en sí parece simple, aunque una primera dificultad ha sido la de conseguir un diccionario de palabras lo bastante grande como para no tener traducciones sincopadas. Una vez hecho esto, ha aparecido el gran problema: las ambigüedades. Una palabra puede tener dos significados y la frase puede no dar pistas acerca de cuál es el verdadero. En una primera aproximación el trance se ha solventado a base de recurrir a la memoria y ver si alguno de los significados aprece previamente. Sin embargo, queda algo mucho más difícil de controlar, la llamada ambigüedad de estructura profunda: dos frases iguales pero con distinto significado. Este problema es el que se intenta resolver y mientras se consigue veremos traductores humanos en las conferencias internacionales.



UNIX (4)

Herramientas para el desarrollo software



ste capítulo está dirigido a aquellos que fundamentalmente utilizan el entorno UNIX para escribir programas. Y en él discutiremos algunas de las herramientas de ayuda que éste ofrece. Como el sistema operativo UNIX fue concebido para el desarrollo, muchas de las herramientas que forman parte del mismo están dirigidos a estas tareas. El sistema UNIX proporciona particularmente un medio rico para la creación de programas y su documentación asociada.

Este capítulo no tiene intención de cubrir con detalle todas las herramientas y lenguajes de programación disponibles. No obstante, nuestra experiencia nos ha mostrado que los usuarios no tienen conocimiento de todas las posibilidades del sistema. Así pues, daremos un repaso a las más importantes: compilador de C., «link-editor», utilidades para la construcción y mantenimiento de librerías y, por último, «make» una aplicación para el mantemiento de programas.

Compilador de C

Cuando en UNIX hablamos de programación, C es el lenguaje que se nos viene a la mente. La historia del lenguaje C y la del sistema operativo UNIX están interrelacionadas de tal forma que podemos casi decir que C se inventó como propuesta para escribir UNIX. Existen, por supuesto, otros lenguajes disponibles, tales como FORTRAN-77 (compilador f77)) y «Rational FORTRAN» (compilador ratfor).

No es nuestro propósito hacer aquí una introducción a este lenguaje, sino más bien crear el entorno en el cual se debe trabajar cuando se usa el sistema operativo UNIX.

Para construir un programa en C, se crea un fichero usando cualquiera de los editores de texto disponibles (ED, VI, SED...) y en él se escribe el código fuente del programa que se quiere ejecutar. Estos ficheros deben crearse con la extensión «c».

La compilación de estos programas se realizan llamando al comando 'cc' (C Compiler) seguido del nombre del mismo. Por ejemplo, para compilar un programa almacenado en el fichero gestor .c el comando sería:

```
$ cc gestor.c
```

Si no existen errores, las diferentes fases de compilación se desarrollarán sin interrupción y aparecerá el fichero llamado «a.out» (Assembler Output), que contiene una versión del programa ejecutable. Si por razones obvias deseamos cambiar el nombre de este fichero, se tienen dos opciones: o bien usamos posteriormente el comando renombrar

```
$ mv a.out gestor
```

El sistema operativo UNIX fue un principio diseñado para trabajos de desarrollo. La mayoría de las herramientas que ofrece están destinadas a este fin.

o bien usamos la opción «o» (minúscula) al compilar:

$ cc - o gestor gestor.c

Una vez obtenido este fichero código, su ejecución es inmediata tecleando «a.out» o el nombre que se le haya asignado (en nuestro caso «gestor»).

Cuando un programa tiene entidad por sí mismo, el proceso de generación de una versión ejecutable es tan simple como lo visto. Pero en la práctica, las cosas no suelen ser tan sencillas. Los programas pueden estar compuestos por varios módulos que hay que compilar separadamente, y posteriormente, combinar en un todo ejecutable.

«Link-Editor»

Como ya se ha mencionado, en la práctica, cualquier proyecto de tamaño razonable tendrá muchos módulos compilados separadamente y cada uno en un fichero distinto. El programa final se generará enlazando cada uno de ellos. El comando 'ld' es el «link-editor» y su función es unir múltiples ficheros de código en un único fichero ejecutable

Cuando en el ejemplo del apartado anterior usábamos el compilador de C, finalizada la compilación propiamente dicha, éste llamaba automáticamente al link-editor para generar el código ejecutable. En el caso que ahora nos ocupa, ésto no interesa, y para que ello no ocurra, ha de realizarse la compilación de los diferentes módulos con la opción '-c'.

$ cc – gestor. c modulo 1.c modulo 2.c

Los ficheros ahora obtenidos tendrán el mismo nombre pero con la extensión 'o' y como vemos en el ejemplo, es posible compilar todos los ficheros de una sola vez. Es importante saber que estos ficheros no son ejecutables; sólo son un resultado parcial antes de obtener el código definitivo.

Seguidamente, usando el comando 'ld' generaremos un fichero 'a.out' que sí será ejecutable, y que podrá modificarse y/o ejecutarse de la forma explicada en el apartado anterior para programas simples.



Los orígenes del sistema operativo UNIX y del lenguaje C están tan interrelacionados que no podemos pensar en uno sin el otro.

Hay un punto que conviene resaltar sobre este comando. En la secuencia de ficheros a enlazar debe colocarse en primer lugar un programa llamado «/lib/ctr0.o» y cuya función es «disparar» la ejecución de otros programas en C. el. El comando 'ld' usa el primer fichero de la secuencia como entrada y si se intenta enlazar los ficheros en otro orden, la ejecución del programa resultante no será posible.

En nuestro ejemplo el comando para enlazar los módulos de gestor sería:

$ ld/lib/crt0.o gestor. o modulo1 o. modulo 2.o

Hay diferentes «disparadores de eje-

cución» para diferentes lenguajes. Por ejemplo el fichero 'lib/fcrt0.o' se usa para programas en FORTRAN, y '/lib/mcrt0.o' cuando los programas usan la librería de punto flotante.

Creación y mantenimiento de librerías con 'ar'

Una de las principales funciones de 'ar' es el mantenimiento de las librerías que los compiladores y montadores utilizan.

En la sección anterior hemos mostrado cómo usar el comando 'ld' para enlazar las diferentes partes de un programa. En la práctica, cuando trabajamos con proyectos complejos, el número de ficheros crece rápidamente y se nos puede escapar de las manos su control.

Usando las posibilidades de link-editor junto con la utilidad 'ar' se pueden mantener todos los módulos necesarios en una librería, y referirnos a ella en la línea de comando 'ld'.

Para comprender mejor el funcionamiento, la explicación se realizará con un ejemplo.

La idea consiste en crear una librería llamada gestor.a que contenga los ficheros «modulo1.o» y «modulo2.o». Y para ello usaremos el comando.

```
$ ar rcv gestor.a. modulo1.o modulo2.o
```

El primer argumento de 'ar' es la opción que nos dice la operación que queremos desarrollar, y que se puede combinar con otras que modifican ligeramente su comportamiento. La opción 'r' significa reemplazar ficheros en la librería, pero en nuestro caso, como estamos creando una nueva y no existen ficheros, simplemente los añade. Las opciones adicionales 'c' y 'v' se utilizan respectivamente para «crear si no existe» y «visualizar las fases».

Si queremos listar los ficheros que gestor.a contiene usaremos el comando:

```
$ ar t. gestor.a
```

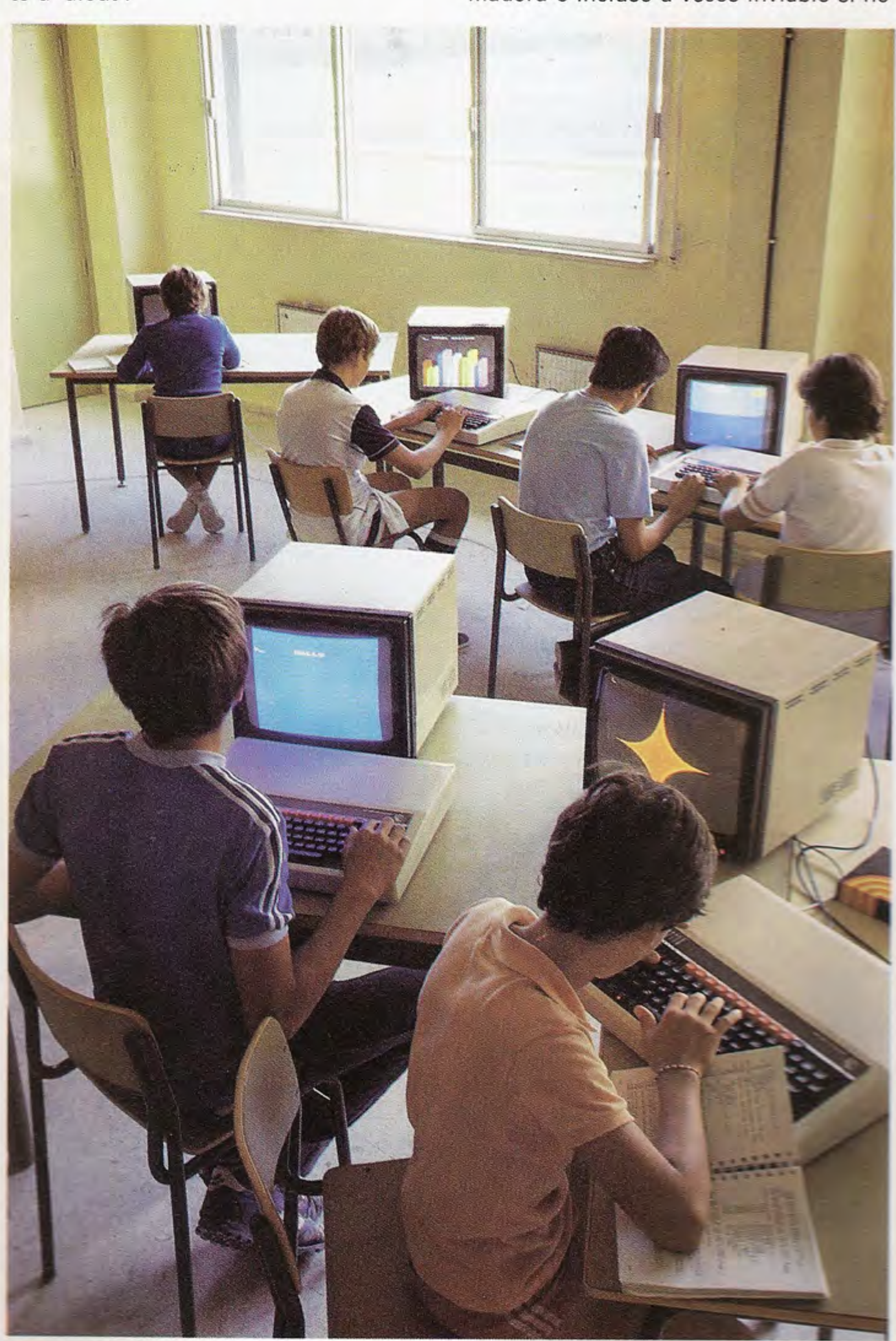
Con la ayuda de esta librería el comando para enlazar el programa gestor quedaría:

```
$ ld -o gestor/ lib/ crt0.o gestor. o gestor.a
```

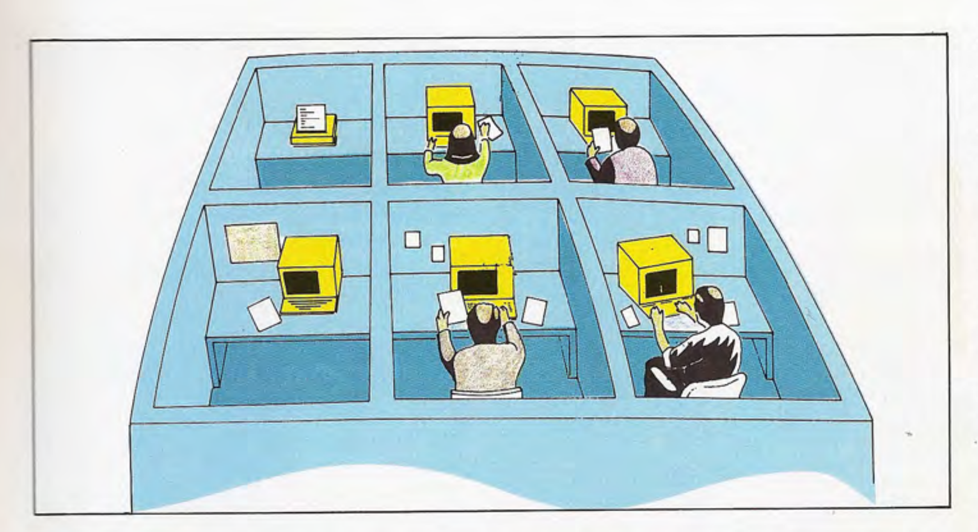
donde la opción '-o' al igual que en el compilador, nos permite bautizar el fichero resultante con un nombre distinto a 'a.out'.

Mantenimiento de programas con 'make'

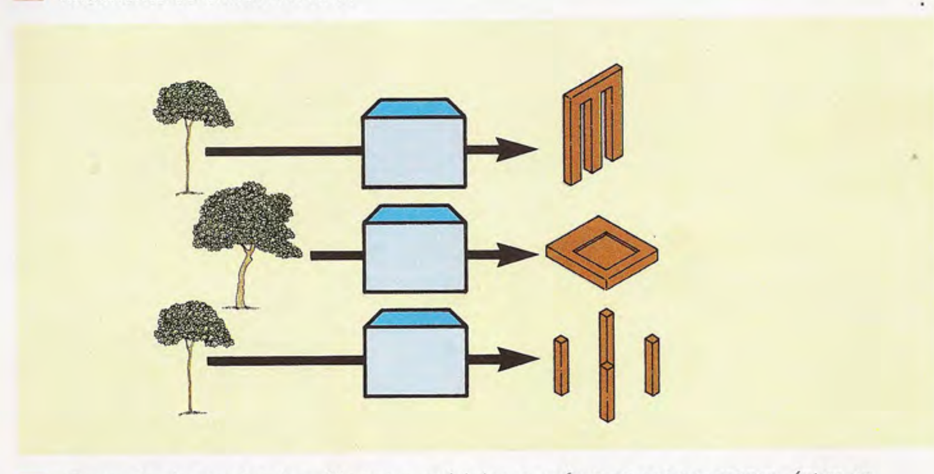
El control de programas complejos es una tarea que puede llegar a ser abrumadora e incluso a veces inviable si no



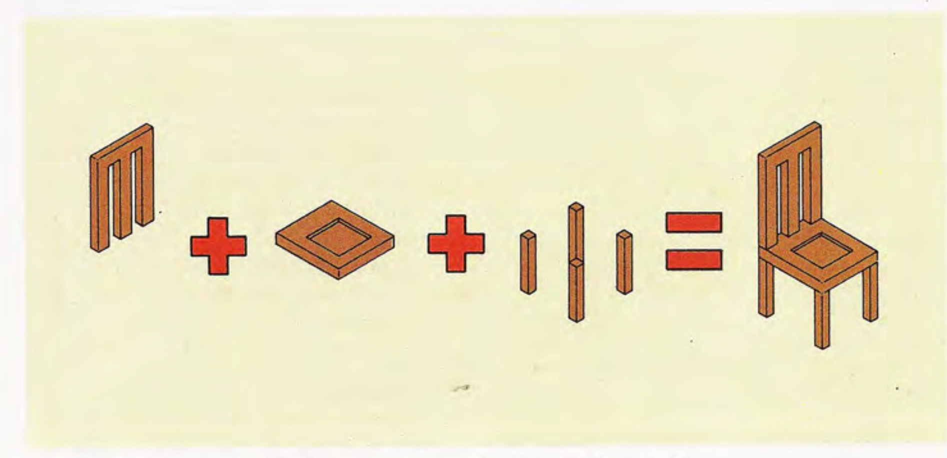
La expansión del sistema operativo UNIX se debe a la facilidad de aprendizaje y a las numerosas herramientas que los estudiantes de la universidad americana desarrollaron.



En los grandes proyectos no tiene sentido hablar de un único programa, sino más bien de un conjunto de módulos desarrollados, independientemente, que finalmente se unirán en uno.



Después de escribir los diferentes módulos que forman un programa, éstos se compilan separadamente con la opción '-c' del compilador.



Los módulos compilados separadamente se unen finalmente en uno mediante el comando 'ld'.

fuera por alguna de las herramientas de ayuda que se ofrecen. Cuando trabajamos en proyectos con grandes cantidades de documentación, pronto ésta nos «desborda» y se nos escapa de las manos.

Make está orientado a facilitar el mantenimiento de programas de ordenador; su trabajo es facilitar el proceso de ir desde la fuente original del programa a su forma ejecutable final. Esto lo hace usando unas reglas descritas previamente que le permiten decidir que comandos ejecutar para llegar a la estructura final deseada.

Ya que make fue originalmente diseñado para mantener programas de ordenador, vamos a mantenernos en este contexto, pero conviene señalar que no es el único.

Todas las decisiones que make toma las hace en función de un fichero de consulta llamado «makefile». En él se sitúan una lista de los ficheros fuente, otra de los ficheros objeto, e información de interdependencias.

En esta sección usaremos nuestro ejemplo de gestor, para entender mejor la forma de usar make.

Antes de nada tenemos que crearnos el fichero makefile que contiene la información para la toma de decisiones de make:

```
$ cat Makefile
FILES = gestor.c. modulo1.c modulo2.c
OBJECTS = gestor. o modulo1.o modu-
lo2.O
        testor: $(OBJECTS)
ld -o gestor / lib/ crto. 0 $(OBJECTS)
```

Es muy corriente llamar a este fichero Makefile o MAKEFILE, con la letra mayúscula M, para que éste aparezca al principio del directorio.

Cualquier línea de la forma:

cadena1 = cadena2

es una macrodefinición. La cadena de la izquierda es el nombre del objeto y la de la derecha es la definición del mismo. Un macro es referenciada precediendo su nombre con el signo $. Si existe más de una cosa en la definición, la referencia debe estar encerrada entra paréntesis o llaves, tal como se muestra en el ejemplo. En este «Makefile» existen dos de estas definiciones, una que referencia a todos los ficheros objetos y otra que referencia a todos los fuente.



Cuando el número de módulos que componen una aplicación excede de un determinado número, empieza a ser engorroso trabajar con todos ellos. Las librerías pueden ser una buena solución.

Después tenemos una línea que especifica que la versión ejecutable del programa gestor depende de los ficheros objeto. Cualquier línea con un nombre seguido de una ':' al principio de la línea, indica una sentencia de dependencias.

Debajo se coloca el comando ld, que es el usuado para generar la versión ejecutable final. Una importante regla de make para makefile es que los comandos del sistema UNIX deben estar precedidos del carácter de tabulación.

Ahora podemos usar el programa make para construir un programa ejecutable de gestor. Bastará con llamar al comando:

```
$ make
```

El cual nos irá mostrando que ocurre mientras se está procesando.

```
cc -c gestor.c
cc -c modulo 1.c
cc -c modulo 2.c
ld -o gestor / lib / crt0.o gestor. o modulo 1. modulo 2.o
```

Si por razones de desarrollo tenemos que modificar alguno de los ficheros, make es lo suficientemente listo como para saber qué acciones tomar para reconstruir «gestor». Sus decisiones se tomarán en función de las interdependencias y de las fechas asociadas a cada uno de los ficheros involucrados en el proyecto.

Así, por ejemplo, si modificamos modulo 1.c al ejecutar make la secuencia de acciones será:

```
$ make
cc -cmodulo 2.c
ld -o gestor / lib/ crt0.o gestor o. modulo 1.modulo 2.o
```

Una opción que suele ser de gran uti-



En grandes proyectos se suele trabajar en grupo. Cada persona trabaja en partes concretas y de forma separada, pero siempre existen una dependencias entre trabajos. Con make el control de interdependencias resulta más sencillo.

lidad es '-n'. Con ella la ejecución de make no se lleva a cabo, pero se nos muestran las operaciones que éste realizaría.

Hay un comando en UNIX muy relacionado con make, 'touch'. Este actualiza posibles ficheros de datos modificados. Por ejemplo, supongamos que queremos asegurarnos de la actualidad de gestor. Si la fecha de modificación de un fichero '.o' es posterior a la de su correspondiente '.c', al ejecutar make, éste no realizaría ninguna operación.

A menudo, cuando trabajamos con grandes proyectos, nos interesa recompilar todos los módulos que lo componen, para asegurar su validez. En estos casos, usando

$ touch *. c

se modifican las fechas de actualización de forma que al usar posteriormente make se realiza el proceso de compilación completo.

Otras herramientas de desarrollo

En capítulos anteriores vimos algunas de las utilidades para procesamiento de ficheros de texto. Estas herramientas pueden sernos de gran utilidad también a la hora de escribir programas.

Aquí estudiaremos algunas de las herramientas disponibles en UNIX de ayuda únicamente a la generación de programas.

La utilidad 'lex' es un miembro más de la familia de herramientas para el procesamiento de texto. 'Lex' es un generador de analizadores lexicográficos. A partir de un fichero de especificaciones, que contiene las reglas del lenguaje, 'lex' obtiene un fichero en código fuente. Este se compila junto con otros programas, y se obtiene un fichero ejecutable llamado 'yylex()' que es el analizador lexicográfico del lenguaje especificado.

Otra utilidad semejante a 'lex' es 'yacc'. Este genera analizadores sintácticos de un lenguaje a partir de las especificaciones gramaticales del mismo. La gramática debe expresarse en BNF, indicando las reglas de producción. Generalmente 'yacc' trabaja en compañía de 'lex' o mejor dicho, del analizador lexicográfico generado por éste. 'yylex()'

Make es una herramienta inteligente que conoce las operaciones que ha de realizar en cada momento en función del estado de los ficheros.

lee la entrada y la divide en señales reconocidas (tokens). Las ventajas que encontramos en 'yacc' son su gran facilidad de modificación y la posibilidad de insertarlo con un conjunto de programas asociados, y los inconvenientes la obtención de un código no eficiente, y la falta de recuperación de errores.

Otra herramienta muy usada es 'awq' y podemos definirla como un filtro de propósito general. En un fichero, que denominaremos 'programa de acciones' se definen una serie de patrones (ristras de símbolos o caracteres) y unas operaciones asociadas a los mismos. 'awq' analizará su entrada buscando los patrones definidos y si encuentra alguno, realizará las acciones asociadas al mismo. Los patrones podrán ser expresiones regulares, expresiones relacionales o una combinación de ambas, y las acciones funciones implícitas a 'awq' o sentencias de shell. Por ejemplo:

?/!Hh¿ello/ p̈rint hola)

La llamada 'awq' se realizará siempre indicando su entrada y su salida, y en caso de que no se haga así, se tomará por defecto la entrada y salida estándar, teclado y pantalla respectivamente. La línea de comando utilizada para esta herramienta tiene un formato:

$ awq programa !files¿

donde program indica el nombre del fichero donde están contenidos los patrones y las operaciones asociadas, y !files¿ indica los ficheros donde ha de buscarse.



Si deseamos asegurar la actualidad de nuestros ficheros objeto, podemos forzar un procesamiento completo de make si previamente usamos Touch.

Indice Temático

MS-DOS (1)

MD-DOS (2)

MS-DOS (3)

OASIS

OS-9

Cuadros

UNIX (4)